(19)日本国特許庁（ＪＰ）　(12)公表特許公報（Ａ）　(11)特許出願公表番号
特表平9－509795
(43)公表日　平成９年(1997)９月30日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	
Ｈ０４Ｎ　1/387		9562－5C	Ｈ０４Ｎ　1/387	
Ｇ０９Ｃ　5/00		7259－5Ｊ	Ｇ０９Ｃ　5/00	
Ｇ１１Ｂ　20/10		7736－5Ｄ	Ｇ１１Ｂ　20/10	Ｈ

審査請求　未請求　予備審査請求　有　（全 86 頁）

(21)出願番号　特願平7－514635
(86)(22)出願日　平成６年(1994)11月16日
(85)翻訳文提出日　平成８年(1996)５月20日
(86)国際出願番号　ＰＣＴ／ＵＳ９４／１３３６６
(87)国際公開番号　ＷＯ９５／１４２８９
(87)国際公開日　平成７年(1995)５月26日
(31)優先権主張番号　１５４，８６６
(32)優先日　1993年11月18日
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　２１５，２８９
(32)優先日　1994年３月17日
(33)優先権主張国　米国（ＵＳ）

(71)出願人　ディジマーク　コーポレイション
アメリカ合衆国　オレゴン州　97223　ポートランド　エス　ダブリュー　グリーンバーグ　ロード　10250　スイート　213
(72)発明者　ローズ　ジェフリー　ビー
アメリカ合衆国　オレゴン州　97068　ウエスト　リン　エス　ダブリュー　チュアラティン　ループ　363
(74)代理人　弁理士　杉村　暁秀　（外１名）

最終頁に続く

(54)【発明の名称】　検証／認証符号化方法および装置

(57)【要約】
検証コード信号を、検証信号を後に識別し、これによってキャリヤを鑑定することが可能な方法において、鑑定すべき（電子データ信号または物理媒体のような）キャリヤ上に加える。本方法および装置は、符号化キャリヤの堅固さの低下と、キャリヤじゅうの検証信号のホログラフィックな拡散とによって特徴付けられる。好適実施例は、検証信号をキャリヤ信号にリアルタイムで埋め込むプロセッサである。

【特許請求の範囲】

１．　後の検証を可能にするように入力信号を検証符号化する方法であって、

ノイズ信号をコード番号によって変調し、署名信号を生成するステップと、

前記入力信号を前記署名信号によって変調し、検証符号化出力信号を生成するステップとを具える方法において、

前記符号化出力信号を分析し、変調に使用した前記コードを識別することができることを特徴とする方法。

２．　符号化出力信号を生成するために入力信号を検証符号化する方法であって、前記入力信号を固有のノイズを有する量子化信号とし、前記信号が聴覚または視覚情報に対応し、前記出力信号の検証符号化が人間の知覚を低下させることなく前記対応する聴覚／視覚情報を保存し、前記検証符号化が前記出力信号の後の検証を可能にし、前記方法がノイズ信号をコード番号によって変調して署名信号を生成し、前記署名信号を前記入力信号に加えて検証符号化出力信号を生成するステップを含み、前記署名信号が入力信号に加えられた場合人間の聴覚／視覚のしきい値より小さい振幅を有し、前記追加のステップが前記署名信号を前記出力信号の全体に渡って分布させることを特徴とする方法。

３．　ディジタル搬送信号を供給することと、前記ディジタル搬送信号を変調し検証信号をごくわずかに埋め込むこととを含むデータ処理方法において、前記変調ディジタル搬送信号を損失データ圧縮によって圧縮して圧縮信号を生成することと、前記圧縮信号を伸長することと、前記伸長信号から前記埋め込まれた検証信号を識別することとによって特徴づけられ、前記損失圧縮が前記埋め込まれた検証信号の再生を妨げないことを特徴とするデータ処理方法。

４．　標本化入力信号を符号化する装置であって、前記標本化入力信号が固有のノイズを有し、前記装置が、入力端子と、ディジタルノイズ源と、検証コードワード用記憶装置と、ポインタを前記検証コードワードのあるビットに保持する手段と、加算器と、出力端子とを含み、前記入力端子を前記加算器の第１入力端子に結合し、前記ノイズ源を前記加算器の第２入力端子に結合し、前記ポ

インタが前記検証コードワードの前記ビットを前記加算器の制御入力端子に供給

し、前記加算器の出力端子を前記出力端子に結合したことを特徴とする装置。

５．　請求の範囲４に記載の装置において、参照表と、第１スケーラと、第２スケーラと、スケール制御装置と、メモリとをさらに含み、前記参照表が前記入力端子に結合された入力端子を有し、前記スケーラの一方が前記参照表の出力端子に結合された制御入力端子を有し、前記スケーラの他方が前記スケール制御装置に結合された制御入力端子を有し、前記スケーラを前記ノイズ源と前記加算器との間に直列に置き、前記メモリが前記ノイズ源と前記加算器の第２入力端子との間の位置に結合された入力端子を有することを特徴とする装置。

６．　固有のノイズを有する標本化入力信号を検証符号化する方法において、

Ｎビットコード番号を供給し、

前記入力信号の複数の標本の各々に対して、

（ａ）時間または空間が変化する変調信号の標本を供給し、

（ｂ）前記Ｎビットコード番号のあるビットを選択し、

（ｃ）前記ビットが第１の値を有する場合、前記変調信号標本を前記入力信号の標本に加算し、検証符号化出力信号の標本を発生することを含むことを特徴とする方法。

７．　請求の範囲６に記載の方法において、前記入力信号の各々の標本に対してステップ（ａ）－（ｃ）を行うことを含むことを特徴とする方法。

８．　請求の範囲６に記載の方法において、前記変調信号を復元することができるデータを、後に使用するために記憶することをさらに含むことを特徴とする方法。

９．　請求の範囲６に記載の方法において、擬似ランダム数を発生し、前記数をスケーリング係数に重み付けし、前記スケーリング係数を前記入力信号標本の関数とすることによって、時間変化する変調信号標本を発生することを含むことを特徴とする方法。

１０．　請求の範囲６に記載の方法において、前記Ｎビットコード番号のあるビットを、前記番号を通じて巡回させ、前記入力信号の各々の連続する標本に対して１ビット位置進ませることによって選択することを含むことを特徴とする

方法。

１１．　請求の範囲６に記載の方法において、前記Ｎビットコード番号の選択されたビットが第２の値を有する場合、前記入力信号の標本から前記変調信号標本を減算し、検証符号化出力信号を発生することを特徴とする方法。

１２．　請求の範囲６に記載の方法に従って処理された信号を記憶した記憶媒体。

１３．　請求の範囲１２に記載の記憶媒体において、磁気媒体としたことを特徴とする記憶媒体。

１４．　請求の範囲１２に記載の記憶媒体において、印刷媒体としたことを特徴とする記憶媒体。

１５．　請求の範囲１２に記載の記憶媒体において、コンパクトディスク（ＣＤ）としたことを特徴とする記憶媒体。

１６．　固有のノイズを有する複数の標本化入力信号の各々を検証符号化する方法において、

前記入力信号の標本を使用してこれらに固有に関係するスケーリング係数を得ることと、

前記スケーリング係数に従って署名基準を重み付けすることと、

前記入力信号の標本を前記重み付けされた署名基準に従って変調することとを行うことを特徴とする方法。

１７．　請求の範囲１６に記載の方法おいて、前記スケーリング係数が、これらが関係する入力信号標本の値とともに単調に増加することを特徴とする方法。

１８．　請求の範囲１６に記載の方法において、標本化入力信号の値における４倍の増加が、これらの関係するスケーリング係数の値における２倍の増加に対応することを特徴とする方法。

１９．　固有のノイズを有する標本化入力信号をＮビット署名ワードによって処理して検証符号化出力信号を発生する方法において、完全なＮビット署名が、１より大きいＭのある値に対してＭ×Ｎ個の標本の長さを有する検証符号化出力信号の引用においてＭ倍という表現を使用することができることを特徴とする方法。

２０．　請求の範囲１９に記載の方法において、前記入力信号の各々の標本を、

前記署名ワードの少なくとも一部に従って処理することを特徴とする方法。

２１．　各々が関連する値を有する多くの要素を含むソース信号を処理する方法おいて、前記ソース信号を埋め込み信号に従って変更して検証コードがこれらの中に符号化されるようにし、前記埋め込み信号および変更された信号の各々が、各々が関係する値を有する多くの要素を含み、前記変更された信号の要素が、前記ソースおよび埋め込み信号の双方における対応する要素と異なった値を有し、前記検証コードおよび特定の擬似ランダム基準データを使用して埋め込み信号を発生し、前記埋め込み信号と検証コードとの間の関連を、前記基準データの有効性なしには識別できないようにしたことを特徴とする方法。

２２．　各々が関連する値を有する多くの要素を含むソース信号を処理する方法おいて、

各々のビットが″１″または″０″値を有するＮビットディジタル検証信号を供給し、

１つが前記ディジタル検証信号における各々のビット位置に関係しているＮ個の異なった基準信号を供給し、

前記検証コードにおける対応するビット位置が″１″値を有する前記基準信号を合計し、それによって埋め込み信号を供給し、

前記埋め込み信号に従って前記ソース信号を変更し、検証コードを前記ソース信号中に符号化するようにし、

前記埋め込まれ変更された信号の各々が関連する値を各々有する多数の要素を含み、前記変更された信号の要素が、前記ソース信号および埋め込み信号の双方における対応する要素と異なった値を有することを特徴とする改良点。

【発明の詳細な説明】

検証／認証符号化方法および装置

発明の分野

本発明は、頑丈な認証コードを電子、光および物理媒体に埋め込むことに関係し、その後、前記媒体の歪みまたは改ざんが起こった後でも、認証目的のこのようなコードの認識することを目的とする。

本発明を、電子画像、シリアルデータ信号（例えば、オーディオおよびビデオ）、感光乳剤フィルムおよび紙幣の検証／認証符号化を含むがこれらに限定されないいくつかの典型的な用途の参照とともに説明する。

発明の背景および要約

″私は、コピーの主人になることによって私の仕事を禁止または変更するいかなる印刷業者または出版業者の支配力を決して受けない″

トマス・ペイン，正義の人，１７９２年

″印刷業者は、彼の出版を許可されたコピーを越えようとはしない″

ミルトン，アエロパゲティカ（Aeropagetica），１６４４年

大昔から、所有権のある資料の許可されない使用および露骨な著作権侵害が、収入の喪失、混乱および芸術的堕落の原因となってきた。

これらの歴史的な問題は、ディジタル技術の登場によって増加してきている。これと共に、許可されない方法において作品をコピーし、これらを再配布する技術は、洗練と、より重要に偏在との新たな高みに到達している。疑わしいコピーとオリジナルとを比較する客観的な手段が無いので、所有者および考えられる訴訟手続きは、前記疑わしいコピーが盗まれたか、許可されない方法において使用されたか否かの主観的判断になってしまう。さらに、作品の元の購入者に対する経路を追跡する簡単な手段が存在せず、考えられる最初に発生した作品の″漏れ

″の場所の追跡に役に立つものが存在しない。

商業作品を保護する種々の方法が試みられてきた。１つは、信号を配布前に符号化方法によってスクランブルし、使用する前に逆スクランブルすることである。しかしながらこの技術は、オリジナル信号と後に逆スクランブルされた信号と

が、これらが横取りされたり記録されたりしないように、閉じられ制御されたネットワークに残っていないことを必要とする。さらにこの装置は、数ドルの追加の費用さえも市場において大きな反応が引き起こされ、信号を知覚するために結局は逆スクランブルしなければならず、したがって容易に記録することができる、オーディオおよびビジュアル作品を多量に取り引きする広範囲な領域においてわずかにしか使用されない。

他の技術の組は、元のオーディオまたはビデオ信号を変調し、電子的な手段によって知覚することがてきるサブリミナル検証信号を含ませることによるものである。このようなシステムの例は、米国特許明細書第４９７２４７１号および欧州特許出願公開明細書第４４１７０２号と、早稲田大学理工学部の研究論文集，No.52 ４５～６０ページ（１９８８）のKomatsu他による、″テレマチックス中に潜伏している画像を使用する認証システム（Authentication System Using Concealed Image in Telematics）″（Komatsuは、この技術に″ディジタル透かし模様″という言葉を使用している）とにおいて見られる。これらの方法に対する基本的な紹介は、１９９３年１１月のByte Magazineの３０９ページの記事″ディジタル署名″において見られる。これらの技術は、ソース作品内の適切に規定されたパターンおよび配列による決定論的信号が検証情報を伝送するという、共通の特徴を有する。特定の用途に関して、これは欠点ではない。しかし一般的に、これは、（ａ）ソース作品の全体を使用しない、（ｂ）決定論的パターンは自称著作権侵害者によって発見および除去される可能性が高い、（ｃ）信号は一般的に′ホログラフィックではない、すなわち検証は、全体の所定の一部のみで行うことが困難であるといった種々の理由から、検証情報を埋め込む非能率的な形態である。（′ホログラフィック′を、ここでは、検証情報が符号化信号じゅうに全体的に分配され、符号化信号の断片の調査からでも完全に識別できるという特性を示すために使用した。この形式の符号化を、ここでは″分配″と呼ぶ。

）

引用した参考文献に共通して、ある文書においては、″．．．情報をある別の目立たない情報中に隠す古い技術″として記述されている、ステガノグラフィ（

steganography）を行ういくつかのプログラムの記述がある。これらのプログラムよって、コンピュータユーザは、かれらのメッセージを、ディジタル画像ファイルおよびディジタルオーディオファイル内にさまざまに隠す。所定のオーディオデータストリームまたはラスタ化画像の最下位ビット（信号データ標本の最小桁ビット）をトグル化することによって行う。これらのプログラムのあるものは、メッセージを最下位ビットに直接埋め込むが、他のものは、始めにメッセージを″前埋め込み″またはスクランブルし、次に暗号化データを最下位ビットに埋め込む。

これらのプログラムの我々が現在理解していることは、これらが一般に、所定のメッセージをそっくりそのまま正確に伝送するために、ディジタルデータのエラーの無い伝送をあてにしていることである。代表的に、メッセージは、１回のみ送信され、すなわち繰り返されない。これらのプログラムは、最下位ビットを完全に″接収″し、実際のデータが消去され、それによってメッセージが配置されるようにも思われる。これは、このようなコードを、所定の画像またはオーディオファイル中のすべてのデータ値の最下位ビットを単に取り除くことによって容易に消去することができることを意味する。これらのおよび他の理由は、我々の発明とステガノグラフィの確立された技術との類似点は、情報をデータファイル中に最小量の知覚可能性によって配置することだけであることを示唆している。埋め込みの特性と埋め込まれた情報の使用とは、これらと異なる。

他の引用した参考文献は、Melenに対する米国特許明細書５３２５１６７号である。所定の文書を認証するサービスにおいて、文書の高解像度走査は、下にある紙それ自体のような文書媒体、またはトナーのような後に用いられた材料に対して、明白に指紋の一種であるパターンおよび″微細な組織構造″を表す。さらにMelenは、この指紋の走査および記憶を後に、目的の文書を走査し、オリジナル文書と比較することによって認証に使用できることを教えている。出願人は、１９９４年２月８日のウォールストリートジャーナルのＢ１ページにおいて報告されているクレジットカード磁気ストライプの極めて高い精度の記録において用いられている同様のアイデアを知っており、このアイデアにおいては、極めて微

細な磁束をあるカードからつぎのカードに固有に向かわせ、これによってクレジットカードの認証を、これらの磁束を予め記録することによって、後に同じであると言われているクレジットカードの記録と比較することによって達成する。

上述した技術の双方は、指紋分析の成熟した科学が基礎を置く同じ検証原理、すなわちある局所的な物理特性の固有のものに基づいていると思われる。これらの方法は、疑わしいものと予め記録したマスタとの″類似″または″相互関係″の１つの判断および／または測定を当てにしている。指紋分析がこれを高い技術にしたとしても、これらの方法は、標本の準備の犯罪に対して無防備であり、Melenkの特許の″抽出″および″スキャナ明細書″は、類似性の判定の結果が片寄る傾向が不可避であり、一致または不一致の信頼性を説明するさらに秘密の″鑑定証明書″を必要とする。本発明の目的は、この鑑定証明書への信頼性を回避し、簡単な俗名″コイン投げ″における一致、すなわち、正確なコイン投げを１６回連続してコールすることができるオッズは何かということに信頼性を置く。指紋、文書、または他のもののの断片を鑑定する試みは、この判定の信頼性の問題を悪化させ、本発明の目的は、直観的な″コイン投げ″信頼性を可能な最も小さい断片に客観的に用いることである。各々すべての文書またはクレジットカード磁気ストライプに関する固有の指紋を記憶することと、後のクロスチェックに容易に利用できるこれらの指紋を有することも、完全に経済的引受けであることを示すべきである。本発明の目的は、ノイズコードの″再使用″と、記憶装置の必要条件を軽減するサービスにおける″雪状画像″とを考慮することである。

Shiang他に対する米国特許明細書第４９２１２７８号は、署名または写真に、訓練されていない眼はノイズと呼ぶであろうが、実際にはモアレパターンと呼ばれるものを拡散させる空間暗号化技術の一種を教えている。本発明のShiangのシステムに対する類似点は、情報を搬送しないノイズ状信号の使用と、クレジットカードおよび他の検証カードにおけるこの原理の使用とであると思われる。

他の引用した特許は、信号または媒体の検証および／または認証のための他の技術に関係する。Hyattに対する米国特許明細書第４９４４０３６号は、″署名

″という言葉を、物理構造を基礎とする固有特性を搬送する信号に等しく用いる

ことができる点以外は本発明には適用できないと思われる。

検証／認証の分野における上述した種々の仕事および他の種々の仕事にもかかわらず、オリジナル信号のコピーとオリジナルとの明確な検証を行う確実で能率的な方法が、依然として必要とされている。望ましくは、この方法は、検証を行うだけでなく、販売時点をより正確にするために、ソースバージョン情報を伝達できるようにすべきである。この方法は、画面上のロゴの配置のような、販売している作品の本質的な品質を妥協すべきではない。この方法は、多数のコピーを行った後でも、および／または信号の圧縮および伸長を行った後でも検証を行えるように堅固にすべきである。この検証方法は、高い消去不可能性または″解読不可能性″を持つべきである。この方法は、オーディオ信号の１０秒の″リフ″またはオリジナル信号の″切り取られ、張り付けられた″小区分のような、オリジナル信号の部分的な断片においても作用できるようにすべきである。

このような方法の存在は、（ａ）作品の許可されない使用の監視と″高速検査″の実行とを費用効果的に行うことができ、（ｂ）この方法が使用されいることとその公表されている結果とが既知である場合、許可されない使用に対する抑止力となることができ、（ｃ）指紋検証と同様に、訴訟において、指紋におけるより可能的に確実に、一致の明白な証拠を提供することができ、著作権侵害者において困難な結果となるであろう。

本発明の好適実施例にしたがって、上述した目的および追加の目的を、微細な検証情報をソース信号全体に埋め込むことによって達成する。好適実施例において、この埋め込みを、ソース信号を僅かなノイズ信号によって符号化された形式に変調することによって達成する。さらに特に、バイナリ検証コードのビット一度に１つ参照し、ソース信号のノイズ信号による変調を制御する。

埋め込みコード信号を有するコピー（″埋め込まれた″コピー）は、販売される作品となり、オリジナルは、安全な場所に保管される。新たなコピーは、最も精密な精査の下でなければオリジナルとほとんど一致し、したがって、商業的価値は妥協されない。新たなコピーが販売され配布された後、多数のコピーによって可能的に歪められても、本明細書は、どのような明細書もオリジナルに対して

明確に鑑定する方法を詳述する。

他の利点の中で、好適実施例の検証信号の使用は、′オリジナル作品のどこか′を単に示すのに対して、大域的（ホログラフィック）であり、擬似自然ノイズ源が検証信号エネルギを最大にする。これは、検証符号化を、像の切断および切り取りのような実世界の数千の堕落したプロセスおよび作品の変形に直面して、より堅固にする。

本発明の上述した特徴および利点と追加の特徴および利点とは、添付した図の参照とともに進める以下の詳細な記述から容易に明らかになるであろう。

図面の簡単な説明

図１は、２つの軸において分離された１次元ディジタル信号の簡単かつ古典的な線図である。

図２は、″微細の″認証信号を他の信号上に埋め込む処理の、ステップの詳細な記述による全体的な概観である。

図３は、オリジナルの疑わしいコピーをどのように検証するかについての漸次の説明である。

図４は、本発明の他の実施例による検証情報によってフィルムを前露光する装置の線図である。

図５は、本発明の″ブラックボックス″実施例の図表である。

図６は、図５の実施例のブロック図である。

図７は、異なったコードワードを有するが同じノイズデータを有する入力データの連続する組を符号化するのに適合した図６の実施例の変形例を示す。

図８は、特有のコード番号を有するビデオテープ製造の各々のフレームを符号化するのに適合した図６の実施例の変形例を示す。

図９Ａ－９Ｃは、本発明のある実施例において使用することができる製造標準ノイズ秒の表示である。

図１０は、標準ノイズコードの検出において使用される集積回路を示す。

図１１は、図１０の実施例において使用することができる標準ノイズコードを検出する処理の流れを示す。

図１２は、本発明の他の実施例による複数の検出器を使用する実施例である。

詳細な説明

説明的な実施例の以下の論考において、言葉″信号″および″画像″を、１、２および２を越える偶数の次元のディジタル信号に言及するのに交換可能に使用する。例を、１次元オーディオ形式ディジタル信号と２次元画像形式ディジタル信号との間で前後に慣例的に切り換える。

本発明の説明的な実施例の詳細を十分に説明するために、最初にディジタル信号の基本的な性質を説明することが必要である。図１は、１次元ディジタル信号の古典的な表現を示す。ｘ軸は、ディジタルの配列″標本″のインデックス番号を規定し、ｙ軸は、ディジタル標本の″２進深度″として規定される有限数のレベルのみにおける存在に抑制されている標本における信号の瞬間的な値である。図１に示す例は、標本値の１６の許可された状態を与える４乗または″４ビット″に対して２の値を有する。

音波のようなオーディオ情報に関して、ディジタル化処理は、連続した現象を時間領域および信号レベル領域の双方において離散的に取り扱うと、一般的に認識されている。そのようなものとして、ディジタル化の処理それ自身が、基本的なエラー原因をもたらし、いずれかの領域における離散的な処理期間より小さい細部を記録することができない。産業界はこれを、時間領域において″エイリアシング″と呼び、信号レベル領域において″量子化ノイズ″と呼ぶ。このように、ディジタル信号の基本エラーフロアが、常に存在する。実効的意味において測定された純粋な量子化ノイズは、１２の平方根を１越えた値か、０．２９ＤＮ程度の値を有することが理論的に既知であり、ここでＤＮは、′ディジタル数′または信号レベルの最も細かい単位増分を表す。例えば、完全な１２ビットディジタイザは、～０．２９ＤＮの固有実効ノイズフロアを伴う、４０９６の許可されたＤＮを有する。

すべての既知の物理測定処理は、連続信号のディジタル形式への変換に追加のノイズを加える。代表的に量子化ノイズは、後に言及するように、直角位相（二乗平均の平方根）において、測定処理の″アナログノイズ″に加わる。

ほとんどすべての商業的および技術的処理によるデシベルスケールの使用は、所定の記録媒体における信号およびノイズの測定として使用される。″信号－ノイズ比″という表現は、一般に、本明細書におけるように使用される。例として、本明細書は、信号ノイズ比を、信号パワーおよびノイズパワーの項として言及し、したがって２０ｄＢは、信号振幅における１０倍の増加を表す。

要約において、本発明の現在の好適な実施例は、全体の信号に、純粋なノイズの形状を有する非常に小さい振幅の符号化信号の付加によってＮビット値を埋め込んだ。通常Ｎを、少なくとも８とし、Ｎビット値の復旧および復号化における最終的な信号－ノイズの考慮によって、より高い限度にする。実際的な問題として、Ｎを、所望の固有の異なった″署名″の数のような、用途の特定の理由に基づいて選択する。説明するために、Ｎ＝１２８とすると、固有のディジタル署名の数は、１０＾＾３８（２＾＾１２８）以上になる。この数は、十分な統計的な確実性をもって作品を検証することと、情報の正確な販売および配布を示すことの双方に対して十分な値以上であると思われる。

この追加の信号の振幅またはパワーは、この方法論を使用する各々すべての用途の、審美的なおよび情報の考慮によって決定する。例えば、非職業的なビデオは、平均的な人間の眼に目立つことなしに、より高い埋め込み信号レベルを有することができるが、高精度オーディオは、″ヒス″における不快な増加を人間の耳が知覚しないように比較的小さい信号レベルのみを採用することができる。これらの供述は、一般的なものであり、各々の用途は、埋め込み信号の信号レベルの選択において、それ自身の基準の組を有する。埋め込み信号のより高いレベルは、より悪質なコピーを検証することができる。他方では、埋め込み信号のより高いレベルは、より不快な知覚されるノイズが、もしかすると配布される作品の価値に影響を及ぼすかもしれない。

本発明の原理を用いることができる異なった用途の範囲を説明するために、本明細書は、２つの異なったシステムを詳述する。第１のもの（よりよい名前が無いために、″バッチ符号化″システムと呼ぶ）は、存在するデータ信号に検証符号化を用いる。第２のもの（よりよい名前が無いために、″リアルタイム符号化″と呼ぶ）は、発生された信号に検証符号化を用いる。これらの当業者は、本発

明の原理を、特に記述したこれらに加えて、多くの他の状況に用いることができることを認識するであろう。

これらの２つのシステムの論考を、どちらの順番で読むこともできる。何人かの読み手は、後者が前者より直観的であることに気づき、他の者にとっては、その反対が真実であろう。

バッチ符号化

実施例の第１の組の以下の論考は、関連する用語を規定する段落によって最も良く始められる。

オリジナル信号を、オリジナルディジタル信号か、非ディジタル信号の高品質にディジタル化されたコピーに適用する。

Ｎビット検証ワードを、８から１２８までのＮレンジを代表的に有し、開示された変換処理を経て最終的にオリジナル信号において配置される検証コードである、固有検証２進値に適用する。示された実施例において、各々のＮビット検証ワードは、値′０１０１′の配列から始まり、疑わしい信号（後述する定義を参照）における信号－ノイズ比の最適化を決定するのに使用される。

Ｎビット検証ワードのm番目のビット値を、Ｎビットワードの左から右に読んだときのm番目の位置に値に対応するゼロまたは１のいずれかとする。例えば、Ｎ＝８検証ワード０１１１０１００の第１（ｍ＝１）ビット値は、値′０′であり、この検証ワードの第２ビット値は、′１′である、等。

m番目の独立埋め込みコード信号を、オリジナル信号に正確に等しい次元および量（例えば、双方が５１２かける５１２ディジタル画像）を有し、（示した実施例においては）ディジタル値の独立した擬似ランダムな配列である信号に適用する。″擬似″は、純粋なランダム状態を哲学的に決定する困難に敬意を払い、″ランダム″信号を発生する種々の許容しうる方法が存在することを示す。いかなる所定のオリジナル信号にも、関係する正確にＮ個の独立した埋め込みコード信号が存在する。

許容しうる知覚されるノイズレベルを、どの位の″余分なノイズ″、すなわち次に記述する複合埋め込みコード信号の振幅を、オリジナル信号に追加し、販売または別な方法の配布に対して許容しうる信号を依然として有していられるかの

用途固有の決定に適用する。本明細書は、許容しうる代表的な値としてノイズにおける１ｄＢの増加を使用するが、これは、全く任意である。

複合埋め込みコード信号は、オリジナル信号と正確に等しい次元および量（例えば、双方が５１２かける５１２ディジタル画像）を有し、Ｎの独立埋め込みコード信号の追加で固有の減衰を含む信号に適用する。独立埋め込みコードを、任意のスケールにおいて発生するが、複合信号の振幅は、前もってセットされた許容しうる知覚られるノイズレベルを越えてはならず、したがってＮの追加独立コード信号の″減衰″を必要とする。

配布可能信号を、オリジナル信号に複合埋め込みコード信号を加えたものから成る、オリジナル信号とほぼ同様のコピーに適用する。これは、外部の社会に配布され、オリジナル信号より僅かに高いが許容しうる″ノイズ特性″を有する信号である。

疑わしい信号を、オリジナルおよび配布された信号の全体的な外観を有し、そのオリジナルに検証が一致する可能性を疑われている信号に適用する。疑わしい信号がＮビット検証ワードに一致する場合、解析すれば分かる。

この第１実施例の詳細な方法論は、Ｎビットワードをｍビット値の各々にこれらの対応する結果として複合信号中に蓄積される独立埋め込みコード信号を乗算することによってオリジナル信号に埋め込むことから始まり、完全に合計された複合信号を次に許容しうる知覚されるノイズ振幅に減衰させ、結果として得られるオリジナル信号に加えられた複合信号が配布可能信号になる。

次にオリジナル信号とＮビット検証ワードとすべてのＮの独立埋め込みコード信号とを、安全な場所に格納する。次に疑わしい信号を見つける。この信号は、多数のコピー、圧縮および伸長、異なった間隔のディジタル信号への再標本化、ディジタルからアナログへそこから戻ってディジタル媒体への変換、またはこれらの項目のなんらかの組み合わせを受けたかもしれない。この信号が、依然としてオリジナルと同様に見える場合、すなわちその本質的な性質が、これらの変換およびノイズの付加のすべてによってまったく破壊されない場合、埋め込み信号のノイズ特性に対する信号に応じて、検証処理を、統計上の確実さのある目的の程度に機能させるべきである。疑わしい信号の改ざんの程度と、オリジナルの許

容しうる知覚されるノイズレベルとを、検証の要求される信頼性レベルの２つのキーパラメータとする。

疑わしい信号における検証処理を、疑わしい信号をディジタルフォーマットおよびオリジナル信号の範囲に再標本化および整列することによって始める。したがって画像が２の因子によって減少している場合、同じ因子によってディジタル的に増大させる必要がある。さらに、音楽の一部が″削除″されているが依然としてオリジナルと同じ標本化率を有する場合、オリジナルのこの削除部分を記録する必要があり、これを代表的に、２つの信号の局所ディジタル相関（通常のディジタル操作）を行い、これの見つけた遅延値を使用して、オリジナルの部分に対する切断部分を記録することによって行う。

疑わしい信号をオリジナルに対して標本化間隔を一致させ記録すると、疑わしい信号の信号レベルを実効的意味においてオリジナルの信号レベルに一致させるべきである。これを、オフセット、２つの信号間のエラーの二乗平均の最小値を前記増幅及びガンマのパラメータの関数として使用することによって最適化されている前記３つのパラメータを探索することによって行うことができる。この点において規格化され記録された、または便利のために単に規格化された疑わしい信号を呼び出すことができる。

このとき新たに適合された対は、規格化された疑わしい信号から減算されたオリジナル信号を有し、差信号を提供する。次に差信号を、Ｎ個の独立埋め込みコード信号と記録されたピーク相関値の各々と相互に関係させる。第１の４ビットコード（′０１０１′）を、０値および１値の平均値と、ノイズ値がより上質の信号を望むなら２つの信号の更なる整合との双方におけるキャリブレータとして使用する（すなわち、０１０１の最適な分離は、２つの信号の最適な整合を示し、Ｎビット検証信号の蓋然的な存在が存在することも示す）。

結果として得られるピーク相関値は、０１０１キャリブレーション配列によって見つけられた０および１の平均値に近接することにより０および１に変換することができる浮動小数点数のノイズの組を形成する。疑わしい信号が本当にオリジナルから得られたものである場合、上述した処理から結果として得られる検証数は、オリジナルのＮビット検証ワードと一致し、″ビットエラー″統計が予測

されたものか既知でないものかを示す。信号－ノイズの考慮は、検証処理においてあの種類の〃ビットエラー〃が存在する場合、検証のＸ％の確率の状態を導くことを決定し、ここでＸは、９９．９％であることが望まれる。疑わしい信号が本当にオリジナルのコピーではない場合、０および１の本質的にランダムの配列が発生し、結果として生じる値の分離の明らかな不足が発生する。すなわち、結果として得られる値をヒストグラムにプロットすると、Ｎビット検証信号の存在は強い２レベル特性を示すが、コードの非存在または異なったオリジナルの異なったコードの存在は、ランダムな正規分布状の形式を示す。このヒストグラムの分離は、検証に対して十分であるが、正確なバイナリ配列を客観的に再生できる場合、検証のより強い証拠となる。

特別な例

カクテルパーティにおける二人の国家首席の高価な絵を手に入れ、この絵が市場においてある妥当な報酬を得るに値するのが確実であるとする。我々は、この絵を売ることを望み、許可されないまたは支払われない方法で使用されないことを保証する。このことと以下のステップとを、図２において要約する。

この絵を、陽画のカラープリントに変換すると仮定する。我々は始めにこれを、代表的な光度測定スペクトル応答曲線を有する通常の高品質白黒スキャナによって、ディジタル化された形式に走査する（カラー画像の３原色の各々において走査することによって、ノイズ比に対してより良い最終的な信号を得ることができるが、このニュアンスは、基本的な処理を記述することに対しては重要ではない）。

ここで、走査された画像は、１２ビットグレイ値または４０９６の許可されたレベルによって規定される精度のグレイスケールを有する４０００×４０００画素のモノクロームディジタル画像になると仮定しよう。我々は、これを、これが前記定義における〃オリジナル信号〃と同一であることを表す〃オリジナルディジタル画像〃と呼ぶ。

走査処理の間、我々は、ディジタル値′３０′に対応する絶対的な黒を任意に設定する。我々は、オリジナルディジタル画像において存在する基本２ディジタル数実効ノイズに加えて、所定の画素の輝度値の平方根の理論上のノイズ（産業

界において″ ショットノイズ″ として知られている）が存在することを見積もる。式において、我々は、

〈ＲＭＳ　Ｎｏｉｓｅ$_{n,m}$〉＝ｓｑｒｔ（４＋（Ｖ$_{n,m}$－３０））　　（１）

を有する。ここで、ｎおよびｍは、画像の行および列において０から３９９９まで変動する簡単な表示値である。Ｓｑｒｔは、平方根である。Ｖは、オリジナルディジタル画像における所定の表示画素のＤＮである。ＲＭＳｎｏｉｓｅの周囲の〈〉括弧は、これが期待される平均値であることを単に意味し、ここで各々すべての画素が、ランダムエラーを個別に有することは明らかである。したがって、ディジタル数または″ 輝度値″ として１２００を有する画素値に対して、我々は、その期待される実効ノイズ値がｓｑｒｔ（１２０４）＝３４．７０であることが分かり、この値は、１２００の平方根である３４．６４にまったく近い。

我々はさらに、画素の固有の輝度値の平方根が、正確に眼が最小の不快なノイズとして知覚する値ではないことを理解しており、したがって我々は、式、

〈RMS Addable Noise$_{n,m}$〉=Ｘ＊sqrt（４＋（Ｖ$_{n,m}$－３０）＾Ｙ）　　（２）

を提案する。ここで、ＸおよびＹを、我々が調節する経験的なパラメータとして加えており、″ ａｄｄａｂｌｅ″ ノイズは、上述した定義による我々の許容しうる知覚されるノイズレベルに属するものである。我々はここで、我々が選択することができるＸおよびＹの正確な値はどの位なのかを実験しようと思うが、我々は、我々が処理の次のステップを実行するのと同時に行う。

我々の処理の次のステップは、我々のＮビット検証ワードのＮを選択することである。我々は、６５５３６の可能な値を有する１６ビット主検証値が、画像が我々のものであることを検証するのに十分に大きく、我々が、我々が追跡を望む画像の１２８のコピーのみを直接販売すると決定し、７ビットに、最初の７ビットの奇数／偶数の加算（すなわち、最初の７ビットにおけるビットのエラー照合）用の第８ビットを加える。ここで必要な全体のビットは、０１０１キャリブレーション配列用４ビットと、主検証用１６ビットと、バージョン用８ビットとであり、我々はここで、最初の２８ビットにおける他のエラー照合値として他の４ビットを投入し、Ｎとして３２ビットを与える。最後の４ビットは、その４ビッ

トを選択するために、多くの業界標準エラー照合方法の１つを使用することができる。

我々はここで、１６ビット主検証数をランダムに決定し、例として、１１０１　０００１　１００１　１１１０を得る。すなわち、販売されたオリジナルの我々の第１のバージョンは、バージョン識別子としてすべて０を有し、エラー照合ビットは一致しなくなる。我々はここで、我々がオリジナルディジタル画像に埋め込む我々の固有３２ビット検証ワードを有する。

これを行うために、我々は、我々の３２ビット検証ワードの各々のビットに対して、３２の独立したランダムの４０００×４０００の符号化画像を発生する。これらのランダム画像を発生する方法を示す。これらを発生する極めて多くの方法が存在する。明らかに最も簡単な方法は、オリジナル写真における走査に使用される同じスキャナにおいて、入力としてこの時だけ黒い画像を置き、次にこれを３２回走査することによってゲインを上昇させることである。この技術の欠点は、大容量のメモリが必要なことと、″固定パターン″ノイズが、各々の独立″ノイズ画像″の一部となることだけである。しかし、固定パターンノイズを、通常の″ダークフレーム″減算技術によって除去することができる。我々は、通常ゲイン設定において２ＤＮ実効ノイズを見つけるよりもむしろ、絶対黒平均値をディジタル数′１００′において設定すると仮定し、ここで我々は、各々すべての画素の平均値について１０ＤＮの実効ノイズを見つける。

我々は次に、中間空間周波数バンドパスフィルタ（空間相乗）を、各々すべての独立ランダム画像に用い、これらから極めて高い空間周波数と極めて低い空間周波数とを本質的に除去する。我々は、幾何学的な歪みや、スキャナにおける汚れや、整合誤りのような簡単な現実世界のエラー源の大部分は、より低い周波数において現れ、我々は、これらの形式の改ざんを回避するために、より高い空間周波数における我々の検証信号に集中したいため、極めて低い周波数を除去する。同様に我々は、所定の画像の多数の世代のコピーや圧縮－伸長変換は、より高い周波数をどんな方法でも破壊する傾向があり、これらの周波数が最も減衰する傾向がある場合、これらの周波数中に多すぎる検証情報が位置する点が存在しな

いようにするために、より高い周波数を除去する。したがって、我々の新たな抽出された独立ノイズ画像は、中央空間周波数によって支配される。実際的な特徴において、我々は我々のスキャナにおいて１２ビット値を使用し、我々はＤＣ値を効果的に除去し、我々の新たな実効ノイズは１０ディジタル数より僅かに少ないことから、これを、結果として得られるランダム画像として－３２から０を通って３１まで変動する６ビット値に圧縮することが有効である。

次に我々は、対応する３２ビット独立検証ワードのビット値において１を有するランダム画像のすべてを互いに加算し、１６ビット署名整数画像における結果を蓄積する。これは、複合埋め込み信号の非減衰および非比例バージョンである。

次に我々は、式２のＸおよびＹパラメータを変化させることによって、複合埋め込み信号をオリジナルディジタル画像に加えることによって視覚的に実験する。式において、我々は、以下の式においてＸの最大化と適切なＹを見つけることとを繰り返し、

$$V_{dist:n,m}=V_{orig:n,m}+V_{comp:n,m}*X*sqrt(4+V_{orig:n,m}\hat{}Y) \qquad (3)$$

ここで、ｄｉｓｔを候補配布可能画像に適用し、すなわち我々は、我々に許容しうる画像を与えるＸおよびＹを見つけることを視覚的に繰り返し、ｏｒｇをオリジナル画像の画素値に適用し、ｃｏｍｐを複合画像の画素値に適用する。ｎおよびｍは、画像の行および列を依然として示し、この操作を４０００×４０００画素の全てにおいて行うことを示す。符号Ｖは、所定の画素および所定の画像のＤＮである。

ここで任意の仮定として、我々は、我々の視覚的実験が、オリジナル画像を候補配布可能画像と比較した場合、Ｘ＝０．０２５およびＹ＝０．６の値が許容しうる値であることを発見したと仮定する。すなわち、″追加ノイズ″を有する配布可能画像は、美的センスにおいてオリジナルに許容しうるほど近い。我々の独立ランダム画像が１０ＤＮ程度のランダム実効ノイズ値を有し、１６程度のこれらの画像を互いに加算することが複合ノイズを４０ＤＮ程度に増加させることから、０．０２５のＸ増加値が、追加の実効ノイズを１ＤＮ程度またはオリジナル

における我々の固有ノイズの振幅の半分に戻すことに注意されたい。これは大雑把に言って、暗い画素値のノイズにおける１ｄＢであり、０．６のＹ値によって変化したより明るい画素においてより高い値に対応するものである。

このようにこれらのＸおよびＹの２つの値によって、我々はここで、我々のオリジナルの配布可能コピーの第１バージョンを構成する。他のバージョンは、単に新たな複合信号を形成し、必要だと考えるならＸを僅かに変更する。我々はここで、オリジナルディジタル画像を、各々のバージョン用３２ビット検証ワードと、３２の独立ランダム４ビット画像と共に固定し、我々のオリジナルの疑わしい著作権侵害の我々の最初のケースを待つ。記憶方法、これは、オリジナル画像用に１４メガバイト程度、ランダム検証埋め込み画像用に３２×０．５バイト×１６０００００＝～２５６メガバイト程度である。これは、１つの高価な画像に関して完全に許容しうる。多少の記憶装置の節約は、簡単な無損失圧縮によって得ることができる。

我々の画像の疑わしい著作権侵害の発見

我々は、我々の画像を販売し、数カ月後、見たところは我々の画像から切り取られ剽窃され、他の様式化された背景場面に置かれたものを見つける。この新たな〝疑わしい〟画像は、所定の雑誌出版の１０００００コピーにおいて印刷されているとする。我々はここで、我々のオリジナル画像の一部が許可されない方法で実際に使用されているかどうかを決定しようとする。図３は、詳細を要約する。

第１のステップは、前記雑誌の発行物を入手し、前記画像をその上に有するページを切り取り、この時、慎重に、しかし慎重に成り過ぎずに、普通の鋏を使用して背景画像から２つの図を切り取ることである。もし可能なら、我々は、２つの図を別々に切り取るよりも、１つの接続された部分のみを切り取る。我々は、これを黒い背景上に張り付け、このことは、視覚的検査を行うのを簡単にする。

我々はここで、我々の安全が保証された場所からオリジナルディジタル画像を３２ビット検証ワードおよび３２の独立埋め込み画像と共に得る。我々は、オリジナルディジタル画像を、標準画像操作ソフトウエアを使用する我々のコンピュ

ータスクリーン上に配置し、我々は、疑わしい画像の我々のマスクされた領域と同じ境界線に沿っておおまかに切断し、同時に同じ様にこの画像をおおまかにマスクする。′おおまか′という言葉を、正確な切断が必要でないことから使用し、これは単に検証統計が合理的に終了されるのを助ける。

次に我々は、マスクされた疑わしい画像を再スケーリングし、我々のマスクされたオリジナルディジタル画像の寸法に大まかに適合させる、すなわち我々は、疑わしい画像を拡大または縮小し、それをオリジナル画像の上に大まかに重ね合わせる。我々がこの大まかな整合を行った後、我々は次に、これらの２つの画像を、自動化されたスケーリングおよび整合プログラムに投入する。このプログラムは、ｘ位置、ｙ位置および空間スケールの３つのパラメータを捜索し、２つの画像間の二乗平均されたエラーが、なんらかの所定のスケール変数とｘおよびｙオフセットとで与えられるという形態の利点を有する。これは、全く標準的な画像処理方法論である。代表的に、これを、大体において滑らかな補完技術を使用して行い、サブ画素精度に行う。探索方法を、多くのものの１つとすることができ、シンプレックス方法を代表的な１つとする。

最適なスケーリングをし、ｘ－ｙ位置変数を見つけたら、次に、前記２つの画像の黒レベルと輝度ゲインとガンマとの最適化における他の探索を行う。再び使用すべき利点の形態は、二乗平均エラーであり、再びシンプレックスまたは他の探索方法論を、これら３つの変数の最適化に使用することができる。これらの３つの変数を最適化した後、我々は、これらの修正を疑わしい画像に用い、それを、オリジナルディジタル画像およびそのマスクの画素間隔およびマスキングとに正確に整合させる。我々はここで、これを基準マスクと呼ぶことができる。

次のステップは、新たに規格化された疑わしい画像から基準マスク領域内のみオリジナルディジタル画像を減算することである。この新たな画像を、差画像と呼ぶ。

次に、３２の独立ランダム埋め込み画像すべてに渡って、マスクされた差画像とマスクされた独立埋め込み画像との間の局所相関を行う。′局所′を、上述した探索手順中、発見された２つの画像の名目上の整合点間のオフセットの＋／－

１画素のオフセット領域によって相関させるのを開始することのみが必要であるという概念に適用する。相関のピークを、０，０オフセットの名目上の整合点に極めて近くすべきであり、我々は、３×３相関値を互いに加算し、我々の３２ビット検証ワードの３２の独立ビットの各々に対する１つの総括的な相関値を与えることができる。

すべての３２ビット位置とこれらの対応するランダム画像のすべてにこれを行った後、我々は、３２値の準浮動小数点配列を有する。最初の４値は、０１０１の我々のキャリブレーション信号を表す。我々はここで、第１および第３浮動少数点値の平均を取り、この浮動小数点値を′０′と呼び、第２および第４値の平均を取り、この浮動少数点値を′１′と呼ぶ。我々は次に、残りのすべての２８ビット値に進み、単にこれらがより近い平均値に基づいて′０′または′１′のいずれかを割り当てる。簡単に言うと、疑わしい画像が実際に我々のオリジナルのコピーの場合、埋め込み３２ビット結果コードは、我々の記録のそれと一致すべきであり、それがコピーでない場合、我々は全体的なランダム状態を得るべきである。３）コピーであるが検証番号と一致しない第３の可能性と、４）コピーではないが適合する第４の可能性があり得る、３）の場合において、処理の信号ノイズ比が重圧を受ける、すなわち疑わしい画像′が正確にオリジナルの極めて粗末なコピーである場合にあり得、４）の場合において、我々が３２ビット検証番号を使用していることから基本的に４０億に１つの可能性がある。我々が４）を本当に心配する場合、我々は、同じ雑誌の異なった刊行物においてこれらのテストを行う第２の独立した試験場を単に有することができる。最後に、これらの値が何を与えるのかを考慮したエラーチェックビットの照合は、処理全体において最終的な出来るかぎり過剰な検査である。ノイズ対する信号が問題に成りうる状況において、これらのエラーチェックビットを、多すぎる害なしに除去することができる。

利益

第１の実施例の完全な説明を、詳細な例によって記述した今、処理ステップとこれらの利点との理論的解釈を指摘することが適切である。

前述の処理の最終的な利益は、検証番号を得ることが、差画像を準備する手段

および方法と完全に独立していることである。すなわち、切断、整合、スケーリング、等のような差画像の準備の方法は、検証番号が存在しない場合、検証番号を発見するオッズが増加せず、真の検証番号が存在する場合、検証処理の信号－ノイズ比のみが役に立つ。検証用画像を準備する方法は、互いに異なっているかもしれず、一致を形成する多数の独立した方法論の可能性を提供する。

オリジナル信号または画像の部分集合において一致を得る能力は、今日の情報に富んだ世界におけるキーポイントである。画像および音声部分の双方の切断および張り付けは、より一般的になり、このような実施例をオリジナル作品が不正に使用されている場合、コピーを検出するのに使用させる。最後に、信号ノイズ比の一致は、コピー作品それ自身がノイズまたは顕著な歪みのいずれかによって顕著に変化している場合のみ困難となり、これらの双方がコピーの商業的価値に影響し、その結果、このシステムを妨げようとすることは、商業的価値における費用の莫大な減少においてのみ行うことができる。

本発明の初期の概念は、１つのみの″スノー状″画像またはランダム信号をオリジナル画像に付加する場合、すなわちＮ＝１の場合であった。この信号を″複合化″することは、この信号の存在または不在における判断を行う（一般的に統計的な）アルゴリズムを使用する、その後の数学的解析を含む。このアプローチを上述した実施例として放棄した理由は、前記信号の存在または不在の検出の確実性において固有の灰色領域が存在することである。″０″から″１″の間で選択する方法を規定する簡単な予め規定されたアルゴリズムと組み合わされた多数のビット段階すなわちＮ＞１への前方への変化によって、本発明は、専門的な統計的解析から、コイン投げのようなランダム２値事象を推定する分野に、確実な問題を変化させた。これは、裁判所および市場の双方における本発明の直観的な許容に関係する有力な特徴として見られる。この全体の問題に対する発明者の考えを要約する類似は、次のようなものである。１つの検証番号の捜索は、コイン投げを１回のみコールし、このコールを行うことを秘密の専門家に期待することに等しいが、本発明の上述したＮ＞１の実施例は、コイン投げをＮ回連続して正確にコールする明白に直観的な原理に期待する。この状況は、非常に苛立たせるものであり、すなわち、画像および音声部分がより小さい範囲を得た場合、１つ

の信号の存在の″改ざん″の問題である。

Ｎ＞１の場合がＮ＝１の実施例よりも好適な実施例である他の理由は、Ｎ＝１の場合において、疑わしい画像を準備し操作する方法が、正の検証を行う可能性を得ることである。したがって、専門家が検証の決定を行う方法は、この決定の必須の部分となる。この決定を行う多数の数学的および統計的アプローチの存在は、いくつかのテストが正の決定を行い、一方他のテストが負の決定を行うという可能性を残し、種々の検証アプローチの相対的な利点についての他の秘密の議論をもたらす。本発明のＮ＞１の好適実施例は、既知の個人コード信号を不正に使用する前処理以外は信号の前処理なしで″コイン投げをＮ回連続してコールする″可能性を増加することができる方法を提供することによって、この他の灰色領域を回避する。

本システムの最も完全な説明は、業界標準および多数の独立したグループが、埋め込み検証番号の適用およびその解読における彼ら自身の手段または″企業内ブランドを設定するようになる場合、見えてくるだろう。多数の独立したグループ検証は、本方法の最終的な目的をさらに強化し、これによって業界標準としての魅力が増強される。

複合埋め込みコード信号の生成における真の極性の使用

上述した論考は、その目的を実行するためにバイナリ技術の０および１の形式論を使用した。特に、Ｎビット検証ワードの０および１は、これらの対応する独立埋め込みコード信号に直接乗算され、複合埋め込みコード信号を形成する（ステップ８、図２）。このアプローチは、その概念の簡単さを確かに有するが、埋め込みコードの記憶と共に埋め込みコード信号の０による乗算は、一種の非能率さを含む。

Ｎビット検証ワードの０および１の性質の形式論を保持するが、これらの対応する埋め込みコード信号を減算させるワードの０を有することが好適である。したがって、図２のステップ８において、Ｎビット検証ワードにおいて′１′に対応する独立埋め込みコード信号を′加算′するだけよりも、Ｎビット検証ワードにおいて′０′に対応する独立埋め込みコード信号の′減算′も行う。

一見して、これは、最終的な複合信号により明白なノイズを付加しているよう

に見える。しかし、０から１へのエネルギ幅分離は増加し、したがって図２のステップ１０において使用される′ゲイン′を、相対して低くすることができる。

我々は、この改良を、真の極性の使用と呼ぶことができる。この改良の主な利点を、′情報の効率′として大きく要約することができる。

独立埋め込みコード信号の′知覚の直交性′

上述した論考は、一般にランダムノイズ状信号を独立埋め込みコード信号として使用することを考察した。これは、発生する信号の恐らく最も簡単な形式である。しかしながら、独立埋め込み信号の組に用いることができる情報最適化の形式が存在し、本出願人は′知覚の直交性′という題目の下に記述する。この用語は、この直交性が、検証情報の信号エネルギを最大化すると同時に、ある知覚しうるしきい値より下に保持すべきであるという現在の追加の要求による、ベクトルの直交性の数学的な概念に大まかに基づいている。他の方法において、埋め込みコード信号は、必然的に現実にランダムであることを必要としない。

感光乳剤ベースの写真の領域における第１実施例の使用および改良

上述した論考は、写真作品に適用できる技術を概説した。以下の節は、この領域の詳細をさらに説明し、これら自身を広範囲な用途に適合させるいくらかの改良を開示する。

論考すべき第１の領域は、ネガフィルム、プリント紙、トランスペアレンシ等のような慣例的な写真作品上に通し番号を前記入または前露光することを含む。一般に、これは、先験的に固有な通し番号（および含有的に所有権およびトラッキング情報）を写真作品中に埋め込む方法である。通し番号それ自体は、余白に追いやられるか、プリントされた写真の背景上にスタンプされるのに対比して、通常の露光された画像の恒久的な部分であり、コピーと別の位置と別の方法とを必要とする。ここで呼ぶ′通し番号′は、一般にＮビット検証ワードと同義語であり、ここでのみ我々は、より一般的な業界用語を使用している。

図２のステップ１１において、本開示は、″オリジナル〔画像〕″をコード画像とともに記憶することを命じる。次に図３のステップ９において、疑わしい画像からオリジナル画像を減算し、これによって、可能な検証コードにノイズおよ

び改ざんが蓄積されたもののすべてを加えたものか残るように命令する。したがって、以前の開示は、複合埋め込み信号なしにオリジナルが存在するという暗黙の仮定をおこなった。

ここで、プリント紙および他のコピー製品を販売する場合において、これは依然としてこの場合、すなわち″オリジナル″が埋め込みコード無しに実際に存在し、第１実施例の基本的な方法論を用いることができる。オリジナルフィルムは、′非符号化オリジナル′として完全に良好に役立つ。

しかしながら、前露光されたフィルムを使用する場合において、複合埋め込み信号がオリジナルフィルム上に予め存在し、したがって予め埋め込まれた信号と分離し、″オリジナル″は、決して存在しない。しかしながら、この後者の場合は、上記で説明した原理をどのように最適に使用するかにおいての観察とともに、ビットをより厳密に調査する（前者の場合は前記で概説した方法に固執する）。

予め番号付けられたネガフィルム、すなわち、各々すべてのフレームに極めて微かな固有複合埋め込み信号を前露光されたネガフィルムの場合の変更の最も明白な点は、以前示した図３のステップ９において現れる。他の相違が確かに存在するが、信号をフィルム上にどの様に何時埋め込むか、コード番号および通し番号をどの様に記憶するか、等のような現実に主として論理的なものである。明らかに、フィルムの前露光は、フィルムの生成および包装の一般的な大量生産工程に大きな変化をもたらす。

図４は、フィルムを前露光する１つの可能性のあるこれ以後の機構の図式的な略図である。′これ以後′を、すべての共通製造工程をすでに行った後に処理を行うことに適用する。結局、経済的規模が、この前露光工程をフィルム製造の連鎖中に直接に配置することを要求する。図４に示すものは、フィルム書き込みシステムとして既知である。コンピュータ１０６は、図２のステップ８において生成される複合信号をその蛍光スクリーン上に表示する。次にフィルムの所定のフレームを、この蛍光スクリーンの像を映すことによって露光し、このときの露光レベルを一般的に極めて微かに、すなわち一般的にごく僅かにする。明らかに、市場が、これをどの位僅かにすべきかの市場自身の要求、すなわち弁護士が見積

もる加えられた″性質″のレベルを設定するであろう。フィルムの各々のフレームを、逐次的に露光し、一般にＣＲＴ１０２において表示される複合画像を各々すべてのフレーム毎に変化させ、これによってフィルムの各々のフレームに異なった通し番号を与える。変換レンズ１０４は、フィルムフレームの焦点変化面とＣＲＴ表面とを強調する。

前露光ネガフィルムの場合における前述の実施例の原理の適用に戻ると、図３のステップ９において、″オリジナル″をその埋め込みコードとともに減算すると、コードがオリジナルの整数部分であることから、明らかにコードも同様に″消去″する。運良く、救済策が存在し、検証を依然として行うことができる。しかしながら、この実施例を改良する技術者は、前露光ネガの場合における検証処理の信号ノイズ比を、非符号化オリジナルが存在する場合の信号ノイズ比に近づけることを要求される。

この問題の簡単な定義は、この点における順番である。疑わしい写真（信号）を仮定した場合、コードがどこかに存在する場合、埋め込み検証コードを見つける。この問題は、上述したようなノイズおよび改ざんの状況内だけでなく、ここでは取り込まれた画像とコードとの間の結合の状況内でも、疑わしい写真内の各々すべての独立埋め込みコード信号の振幅の発見の１つに減少する。′結合′を、ここでは取り込まれた画像が相関に″ランダムにバイアスする″という概念に適用する。

このように、信号結合のこの追加の項目を心に止めておくと、検証処理は、各々すべての独立埋め込みコードの信号振幅を見積もる（図３のステップ１２では相関結果を得るのに対して）。我々の検証コードが疑わしい写真中に存在する場合、発見される振幅は、′１′を割り当てられている正振幅と′０′を割り当てられている負振幅を有する両振幅に分割されている。我々の固有検証コードは、それ自身を明らかにする。他方で、このような検証コードが存在しない場合、または何か他のコードである場合、振幅のランダムガウス状分布は、値のランダムな寄せ集めによって見つかる。

独立埋め込みコードの振幅をどの様に発見するかについてのいくつかの更なる

詳細を与えることが残っている。再び、運良く、この厳密な問題は、他の技術上の用途において処理されている。さらに、この問題と少しの食料とを数学者と統

計学者とで混み合っているいる部屋に投げ込めば、ある適当な期間の後、半ダースの最適化された方法論が必ず出で来るであろう。それは、ある程度きれいに定義された問題である。

ある特別な例としての解決法は、天文学上の撮像の分野から生じる。ここで、成熟した先行技術は、〝熱ノイズフレーム〟を物体の所定のＣＣＤ画像から減算する。しかしながらしばしば、熱フレームの減算においてどの位のスケーリング係数を使用するのかは明確に既知ではなく、正確なスケーリング係数の探索が行われる。これは、明確に本実施例のこのステップの仕事である。

一般的な習慣は、単に一般的な探索アルゴリズムをスケーリング係数において行い、スケーリング係数を選択し、新たな画像を、

新たな画像＝獲得された画像－スケーリング係数×熱画像　　（４）

によって形成する。

新たな画像に高速フーリエ変換ルーチンを用い、最終的に、新たな画像の積分高周波内容を最小化するスケール係数を見つける。この個々の量の最小化による一般的な形式の探索操作は、非常に一般的である。したがって発見されたスケール係数は、探索された〝振幅〟である。考察されているがまだ実現されていない改良は、獲得された画像のより高い導関数と埋め込みコードとの結合を、見積もり、計算されたスケール係数から除去することである。すなわち、上述した結合による特定のバイアス効果が存在し、最終的には理論上および経験的な実験の双方によって明らかにされ除去されるべきである。

信号または画像の変化の検出における使用および改良

全体として信号または画像を検証することの基本的な必要性から離れて、信号または画像に対して起こりうる変化を検出する多少偏在する必要性も存在する。以下の節は、前記実施例を、特定の変更および改良によって、この領域における有力な道具としてどのように使用することができるかを記述する。

最初に要約するために、我々は、前記で概説した基本的な方法を使用して正に

検証された所定の信号または画像を有すると仮定する。すなわち、我々は、その

Ｎビット検証ワードと、その独立埋め込みコード信号と、その複合埋め込みコードとを知っている。次に我々は、我々の所定の信号または画像内の複合コードの振幅の空間マップを全く簡単に形成することができる。さらに我々は、規格化マップ、すなわちある大域的平均値の周囲を変化するマップを与えるために、この振幅マップを既知の複合コードの空間振幅によって分割することができる。このマップの簡単な調査によって、我々は、明白に変化して、規格化振幅の値が代表的なノイズおよび改ざん（エラー）に単に基づくしきい値のある統計上の組より低下するどの様な領域も、視覚的に検出することができる。

振幅マップの形成の実施の詳細は、種々の選択を有する。１つは、上述した信号振幅の決定に使用したのと同じ手順を行うことであり、ここでは我々は、我々が調査している領域付近に中心が位置する正規重み関数を信号／画像のすべての所定の領域に乗算する。

万能コード対カスタムコード

本明細書は、ここまでは、各々すべてのソース信号が独立埋め込みコード信号の自分自身の組をどのように有するのかを概説した。これは、オリジナルに加えて相当な量の追加のコード情報の記憶を必要とし、多くの用途には、より経済的な形式が適切であろう。

あるこのような節約のためのアプローチは、一組のソース作品に共通の独立埋め込みコード信号の所定の組を有することである。例えば、我々の１０００枚の画像がすべて、独立埋め込みコード信号の同じ基本的な組を利用することができる。このときこれらのコードに必要とされる記憶は、ソース作品に必要とされる記憶全体のほんの一部となる。

さらに、いくつかの用途は、独立埋め込みコード信号の万能組、すなわち配布された作品のすべての場合に同一のままであるコードを利用することができる。この形式に必要なものは、Ｎビット検証ワードそれ自身を隠そうとし、このワードを読み取ることができる統一された装置を有するシステムによって分かるであろう。これを、読み取り位置の点において判断する／しないシステムにおいて使

用することができる。この設定をする潜在的な欠点は、万能コードは、より追跡または盗難されやすく、したがってこれらは、前記で開示した設備の装置および

方法論より安全ではない。恐らくこれは、′高い安全性′と′気密の安全性′との間の差であり、潜在的な用途の大部分にとってはあまり重要でない区別である。大域埋め込みコードを付けることができる紙、文書、プラスチック加工身分証明カード、および他の材料への印刷における使用

用語′信号′を、ディジタルデータ情報、オーディオ信号、画像、等を指示するためにしばしば狭義において使用する。′信号′のより広義の解釈と、より一般的に意図されたものとは、どの様な材料のどの様な形式の変化も含む。したがって、一般的な紙の断片のマイクロトポロギーは、信号（例えばｘ－ｙ座標の関数としての高さ）となる。プラスチックの平坦な断片の屈折特性は、（空間の関数としての）信号となる。要点は、写真感光乳剤、オーディオ信号、およびディジタル化情報は、本発明の原理を使用することができる信号の唯一の形式ではないということである。

適切な場合として、ブライユ点字印刷機械に大変よく似た機械を、前記で概説した固有の′ノイズ状′検証を付けるように設計することができる。これらの検証を、ブライユ点字の形成において加えられるよりはるかに小さい圧力によって、そのパターンが書類の普通の使用者によって認められないような位置に加えることができる。しかし、本明細書のステップを続け、微細な検証の機構によってこれらを用いることによって、固有検証コードを、日常の便箋としての目的を意図したものや、重要な文書、法的な提出物、または他の保証された作品である、どのような紙面にも配置することができる。

このような実施例における検証作品の読み取りは、一般的に、文書を光学的に種々の角度において単に読み取ることによって行われる。これは、紙面のマイクロトポロギを推論するために安価な方法となる。確かに紙のトポロギを読み取る他の形式も可能である。

例えば運転免許書である身分証明カードのようなプラスチックに封入された作品の場合において、同様のブライユ点字印刷機械に類似した機械を、固有検証コ

ードを付けるのに利用することができる。感光材料の薄い層をプラスチックの内側に埋め込み、′感光′させることもできる。

′ノイズ状′信号によって変調させることができる材料が存在するところならどこでも、この材料は、固有検証コードおよび本発明の原理を利用するための適切なキャリヤとなることは明らかである。経済的に検証情報を付加し、信号レベルを各々すべての用途がそれ自身に対して規定する許容しうるしきい値より下に保持する問題が残りの全てである。

付録Ａの説明

付録Ａは、８ビット白黒画像システムのための前述の実施例の実現および証明のソースコードを含む。

リアルタイムエンコーダ

実施例の第１の組は、画像または信号の符号化を行う標準的なマイクロプロセッサまたはコンピュータを最も一般に使用し、代表的なフォンノイマン型プロセッサより速くすることができるカスタム符号化装置を使用することができる。このようなシステムを、すべての様式のシリアルデータストリームに使用することができる。

音楽およびビデオテープ記録を、シリアルデータストリーム、しばしば著作権侵害を受けるデータストリームの例とする。許可された記録を検証データによって符号化し、著作権侵害された盗品をこれらが形成されたものからオリジナルを探索できるようにしたならば、実施の試みの助けとなるであろう。

著作権侵害は、本発明を必要とすることの１つにすぎない。他の事は、認証である。しばしば、データの所定の組が（しばしばその発生から数年後）実際に何を意図しているのかを確認することが重要になる。

これらおよび他の必要性を説明するために、

図５のシステム２００を使用することができる。システム２００を、検証符号化ブラックボックス２０２として考えることができる。システム２００は、（後に″マスタ″または″非符号化″信号と呼ばれる）入力信号およびコードワードを受け、検証符号化出力信号を（一般にリアルタイムで）発生する。（通常、本シ

ステムは、後の復号化に使用するキーデータを提供する。）

〝ブラックボックス〟２０２の中身は、種々の形態をとることができる。典型的なブラックボックスシステムを図６に示し、これは、参照表２０４と、ディジタルノイズ源２０６と、第１および第２スケーラ２０８および２１０と、加算器／減算器２１２と、メモリ２１４と、レジスタ２１６とを含む。

（図示した実施例においては、１０００００００標本毎秒のレートにおいて供給される８－２０ビットデータ信号であるが、他の実施例においては、適切なＡ／ＤおよびＤ／Ａコンバータが設けられている場合、アナログ信号とすることができる）入力信号を、入力端子２１８から参照表２０４のアドレス入力端子２２０に供給する。各々の入力標本（すなわち、参照表アドレス）に対して、参照表は、対応する８ビットディジタル出力ワードを供給する。この出力ワードを、第１スケーラ２０８の第１入力端子に供給されるスケーリング係数として使用する。

第１スケーラ２０８は、第２入力端子を有し、この入力端子にノイズ源２０６から８ビットディジタルノイズ信号を供給する。（図示した実施例において、ノイズ源２０６は、アナログノイズ源２２２およびアナログーディジタルコンバータ２２４を具えるが、再び、他の手段を使用することができる。）図示した実施例におけるノイズ源は、５０から１００のディジタル数（例えば、－７５から＋７５）の半値全幅（ＦＷＨＭ）を有する、ゼロ平均出力値を有する。

第１スケーラ２０８は、その入力端子における２つの８ビットワード（スケール係数およびノイズ）を乗算し、システム入力信号の各々の標本に対して、１つの１６ビット出力ワードを発生する。ノイズ信号がゼロ平均値を有することから、第１スケーラの出力信号も同様にゼロ平均値を有する。

第１スケーラ２０８の出力信号を、第２スケーラ２１０の入力端子に供給する。第２スケーラは、大域的スケーリング機能を行い、最終的に入力データ信号中に埋め込まれる検証信号の絶対量を確立する。前記スケーリング係数を、スケール制御装置２２６（簡単な加減抵抗器から、グラフィカルユーザインタフェースにおいて図式的に実現された制御まで、多くの形態をとることができる）によって設定し、別個の用途の要求にしたがって変更すべきこの係数を可能にする。第

２スケーラ２１０は、その出力ライン２２８にスケールノイズ信号を発生する。このスケールノイズ信号の各々の標本を、メモリ２１４に順次記憶する。

（図示した実施例において、第１スケーラ２０８からの出力信号は、－１５００と＋１５００（１０進数）との間で変化しうるが、第２スケーラ２１０からの出力信号は、小さい１つの数字である（－２と＋２との間のような）。）

レジスタ２１６は、多ビット検証コードワードを記憶する。図示した実施例において、このコードワードは、８ビットから成るが、より大きいコードワード（数１００ビットに及ぶ）が一般的に使用される。これらのビットを一度に１つ参照し、入力信号のスケールノイズ信号による変調の程度を制御する。

特に、ポインタ２３０を、レジスタ２１６におけるコードワードのビット位置を通じて順次に循環させ、″０″または″１″の制御ビットを加算器／減算器２１２の制御入力端子２３２に供給する。ある入力信号標本に関して、制御ビットが″１″の場合、ライン２３２におけるスケールノイズ信号標本を入力信号標本に加算する。制御ビットが″０″の場合、スケールノイズ信号標本を入力信号標本から減算する。加算器／減算器２１２からの出力端子は、ブラックボックスの出力信号を発生する。

コードワードのビットに従ったスケールノイズ信号の加算または減算は、一般にごく僅かな入力信号の変調に影響する。しかしながら、メモリ２１４の内容の認識によって、ユーザは、符号化を後に復号化し、オリジナル符号化処理において使用されるコード番号を決定することができる。（実際に、メモリ２１４の使用は、以下に説明するように任意である。）

符号化信号を、印刷された画像に変換された形式、磁気媒体（フロッピーディスク、アナログまたはＤＡＴテープ、等）に記憶された形式、ＣＤ－ＲＯＭ、等々を含むよく知られた方法において配布することができることが認識されるだろう。

復号化

種々の技術を、疑わしい信号が符号化されているままで検証コードを決定するのに使用することができる。２つを以下で論考する。第１のものは、多くの用途

にとって後者よりも好適ではないが、ここで論考することによって、読み手は、本発明を理解するより完全な状況を得るであろう。

さらに特に、第１の復号化方法は、差方法であり、オリジナル信号の対応する標本を疑わしい信号から減算し、差標本を得ることによるものであり、次に決定論的に符号化された証印（すなわち、記憶されたノイズデータ）に対して調査する。したがってこのアプローチを、″標本に基づく決定論的″復号化技術と呼ぶことができる。

第２の復号化技術は、オリジナル信号を使用しない。個々の標本を調査して、予め決められたノイズ特性を探すこともしない。むしろ、疑わしい信号の統計値（またはこれらの一部）を、全体として考え、分析して、信号全体に充満する検証信号の存在を識別する。充満に対する言及は、検証コード全体を、疑わしい信号の小さい部分から識別することができることを意味する。したがってこの後者のアプローチを、″ホログラフィック統計的″復号化技術と呼ぶことができる。

これらの方法の双方は、疑わしい信号をオリジナルに整合させることによって開始する。これは、スケーリング（例えば、振幅、継続時間、色バランス、等における）と、オリジナルの標本化レートを復旧するための標本化（または再標本化）とを必要とする。上述した実施例におけるように、この整合機能に関係する操作を行うことができる種々の良く理解された技術が存在する。

言及したように、第１の復号化アプローチは、オリジナル信号を整合された疑わしい信号から減算し、差信号を残すことによって生じる。次に連続する差信号標本の極性を、対応する記憶されたノイズ標本信号の極性と比較し、検証コードを決定することができる。すなわち、第１差信号標本の極性が第１ノイズ信号標本の極性と一致した場合、検出コードの第１ビットを″１″とする。（このような場合、９番目、１７番目、２５番目、等の標本の極性も、すべて正とすべきである。）第１差信号標本の極性が、対応するノイズ信号標本の極性と反対である場合、検証コードの第１ビットを″０″とする。

差信号の８つの連続する標本について前述の分析を行うことによって、オリジナルコードワードを具えるビットの配列を決定することができる。好適実施例に

おけるように、符号化中、ポインタ２３０が、コードワードを通じて一度に１ビット進み、第１ビットによって開始する場合、差信号の最初の８つの標本を分析し、８ビットコードワードの値を唯一決定することができる。

ノイズの無い世界（ここで言っているノイズは、検証符号化に作用するノイズと無関係である）において、前述の分析は、常に正確な検証コードをもたらす。しかし、ノイズの無い世界においてのみ適合した処理は、実際は利用が制限される。

（さらに、ノイズの無い状況における信号の正確な検証を、種々の他のより簡単な方法、例えば、チェックサム、すなわち、疑わしい信号およびオリジナル信号間の統計的不可能性一致、等によって処理することができる。）

復号化においてノイズが引き起こす異常を、信号の大きな部分を分析することによって、ある程度まで、処理することができるが、このような異常は、処理の信頼性において実際的な上限を依然として設定する。さらに、直面しなければならない悪人は、常にランダムノイズより優しくない。むしろ、人間によって引き起こされる形式の改ざん、歪み、不正な操作、等が、益々選択される。これらのような場合において、検証の信頼性の所望の程度は、他のアプローチによってのみ達成される。

現在好適なアプローチ（″ホログラフィック、統計的″復号化技術）は、疑わしい信号を特定のノイズデータ（代表的に、メモリ２１４に記憶されたデータ）と再結合し、結果として得られる信号のエントロピを分析することに頼っている。″エントロピ″を、その最も厳密な数学的定義において理解する必要はなく、単に、ランダム性（ノイズ、平坦性、雪状性、等）を記述する最も簡潔な言葉とする。

大部分のシリアルデータ信号は、ランダムではない。すなわちある標本は、通常、隣接する標本と、ある程度相関する。対照的に、代表的にノイズは、ランダムである。ランダム信号（例えば、ノイズ）を、非ランダム信号に加算した場合（またはこれから減算した場合）、結果として得られる信号のエントロピは、一般的に増加する。すなわち、結果として得られる信号は、元の信号よりもランダ

ムな偏差を有する。これは、現在の符号化処理によって発生された符号化出力信号の場合であり、元の非符号化信号より大きいエントロピを有する。

対照的に、ランダム信号の非ランダム信号への加算（またはこれからの減算）が、エントロピを減少させる場合、なんからの例外が発生する。好適な復号化処理を使用し、埋め込み検証コードを検出することが、この例外である。

このエントロピに基づく復号化方法を十分に理解するために、８番目毎に同様の処理であるオリジナル復号化処理の特徴を強調することが第１の助けとなる。

前記で論考した符号化処理において、ポインタ２３０は、コードワードを通じて、入力信号の各々の連続する標本毎に１ビット増分する。コードワードが８ビット長の場合、ポインタは、コードワード中の同じビット位置に８番目の標本毎に戻ってくる。このビットが″１″ならば、入力信号にノイズを加算し、このビットが″０″ならば、入力信号からノイズを減算する。したがってポインタ２３０の周期的な進行によって、符号化信号の８番目毎の標本は、特徴を共有し、ポインタ２３０によってアドレスされているコードワードのビットが″１″か″０″に応じて、これらをすべて、対応するノイズデータによって増加するか（反対でもよい）、これらをすべて減少する。

この特徴を利用するために、エントロピに基づく復号化処理は、疑わしい信号の８ビット毎に、同様の方法で処理する。特に、疑わしい信号の１番目、９番目、１７番目、２５番目、等の標本に、メモリ２１４に記憶された対応するスケールのノイズ信号（すなわち、各々、１番目、９番目、１７番目、２５番目、等のメモリ位置に記憶されたノイズ信号）を加算することによって、処理は開始する。次に、結果として得られる信号（すなわち、８番目の標本毎に変更された疑わしい信号）のエントロピを計算する。

（信号のエントロピまたはランダム性の計算法は、当業者には良く知られている。一般的に受け入れられているものは、各々の標本点において信号の導関数を取り、これらの値を二乗し、信号全体に渡って合計することである。）

次に、上記のステップを繰り返し、この時、記憶されたノイズ値を、疑わしい信号の１番目、９番目、１７番目、２５番目、等の標本から減算する。

これらの２つの操作の一方は、符号化処理を取消し、結果として得られる信号のエントロピを減少させ、他方は、それを悪化させる。メモリ２１４中のノイズデータの疑わしい信号への加算が、そのエントロピを減少させる場合、このデータは、以前オリジナル信号から減算されたに違いない。これは、ポインタ２３０が、これらの標本が符号化された時、″０″ビットを指していたことを示す。（加算器／減算器２１２の制御入力端子における″０″は、スケールノイズの入力信号からの減算を生じる。）

反対に、ノイズデータの疑わしい信号の８番目毎の標本からの減算が、そのエントロピを減少させる場合、符号化処理は、以前このノイズを加算したに違いない。これは、ポインタ２３０が、標本１、９、１７、２５、等が符号化された時、″１″を指していたことを示す。エントロピの減少が、記憶されたノイズデータの疑わしい信号への／からの（ａ）加算または（ｂ）減算のいずれかによるものかに注目することによって、コードワードの第１ビットが、（ａ）″０″または（ｂ）″１″であるかを決定することができる。

上記の操作を、疑わしい信号の第２標本（すなわち、２、１０）１８、２６．．．）に始まる一定の間隔をおいた標本のグループに対して行う。結果として得られる信号のエントロピは、コードワードの第２ビットが、″０″または″１″のいずれであるかを示す。疑わしい信号の続く６個のグループに対して同様に、コードワードの８ビットすべてを識別するまで繰り返す。

上述したアプローチが、個々の標本の値を変更する改ざん機構に変動されないことは、理解されるであろう。すなわち、代わりに、このアプローチは、信号のエントロピを、結果における高い程度の信頼性を生じるものと見なす。さらに、信号のわずかな抜粋をこの方法によって分析し、オリジナル著作物の細部の著作権侵害も検出することができる。したがって結果として、疑わしい信号の自然および人的の改ざんの双方に直面して、統計的に健全である。

さらに、このリアルタイムの実施例におけるＮビットコードワードの使用が、バッチ符号化システムに関連して、上述したのと類似の利益をもたらすことが理解されるだろう。（実際は、本実施例を、バッチ符号化システムにおいて、Ｎ個

の差ノイズ信号を使用するものとして概念化することができる。第１ノイズ信号を、入力信号と同じ広がりを有し、標本間に０を有する１番目、９番目、１７番目、２５番目、等の標本（Ｎ＝８として）におけるスケールノイズ信号を具える信号とする。第２ノイズ信号を、標本間に０を有する２番目、１０番目、１８番目、２６番目、等の標本におけるスケールノイズ信号を具える同様の信号とする。その他同様。これらの信号をすべて混合し、複合ノイズ信号を発生する。）このようなシステムにおいて固有の重要な利点の１つは、一致が真に一致である統計的な信頼性（検証コードの各々の連続するビットとともに倍になる信頼性）の程度が高いことである。このシステムは、疑わしい信号の１つの決定論的な埋め込みコード信号に対する主観的な評価に頼らない。

説明的な変形例

上述した説明から、示したシステムに対して、基本的な原理を変更することなく、多くの変更を行えることが認識されるだろう。これらの変形例のいくつかを、以下に記述する。

上述した復号化処理は、どちらの操作がエントロピを減少させるのかを見つけるために、記憶されたノイズデータの疑わしい信号への／からの加算および減算の双方を試す。他の実施例において、これらの操作の一方のみを行う必要がある。例えば、ある一方の復号化処理において、疑わしい信号の８番目毎の標本に対応する記憶されたノイズデータを、前記標本に加算のみ行う。結果として得られる信号がそのために増加した場合、コードワードの対応するビットは、″１″である（すなわち、このノイズは、以前、復号化処理中に加算されており、再び加算されたために、信号のランダム性のみが増加した）。結果として得られる信号がそのために減少した場合、コードワードの対応するビットは、″０″である。記憶されたノイズ信号を減算するエントロピの他の試験は、必要ない。

検証処理（符号化および復号化）の統計的信頼性を、大域的スケーリングファクタの適切な選択によって、どのような信頼性しきい値（例えば、９９．９％、９９．９９％、９９．９９９％、等）も実質的に越えるように設計することができる。なんらかの所定の用途（大部分の用途においては必要ない）における特別

の信頼性を、復号化処理を再検査することによって達成することができる。

復号化処理を再検査する一つの方法は、識別されたコードワードのビットに従って疑わしい信号から記憶されたノイズデータを除去し、″復旧″信号を発生する（例えば、コードワードの第１ビットが″１″であることが分かった場合、メモリ２１４の第１、第９、第１７、等の位置に記憶されたノイズ標本を、疑わしい信号の対応する標本から減算する）ことである。記憶されたノイズ信号のエントロピを測定し、他の測定における基線として使用する。次に、この処理を繰り返し、この時、変更されたコードワードに従って、記憶されたノイズデータを疑わしい信号から除去する。変更されたコードワードは、結合された（例えば、第１）１ビットを除いて、識別されたコードワードと同一である。結果として得られる信号のエントロピを測定し、前記基線と比較する。識別されたコードワード

におけるビットのトグリングが増加されたエントロピを生じる場合、識別されたコードワードのそのビットの精度は、確実になる。トグルされた確認されたコードワードの異なったビット毎に、コードワードのすべてのビットが検査されるまで、この処理を繰り返す。各々の変更の結果として、基線値に比べてエントロピが増加する。

メモリ２１４に記憶されたデータは、種々の二者択一を受ける。上述した論考において、メモリ２１４は、スケールノイズデータを含む。他の実施例において、非スケールノイズデータを、代わりに記憶することができる。

さらに他の実施例において、入力信号それ自身の少なくとも一部を、メモリ２１４に記憶することが望ましいかもしれない。例えば、このメモリは、８つの署名ビットをノイズ標本に割り当て、１６ビットを１８または２０ビットオーディオ信号標本の最上位ビットの記憶に割り当てることができる。これは、いくつかの利益を有する。１つは、″疑わしい″信号の整合が簡単になることである。他の利益は、既に符号化された入力信号を符号化する場合、メモリ２１４内のデータを、どちらの符号化処理が最初に行われたかを識別するのに使用することができることである。すなわち、メモリ２１４内の入力信号データから（不十分にもかかわらず）、一般に、２つのコードワードのどちらが符号化されているかを決

定することができる。

メモリ２１４のさらに他の二者択一は、全体を省略できることである。

これを達成できる方法の１つは、符号化処理において、既知の鍵番号によって種を蒔かれるアルゴリズム式ノイズ源のような決定論的ノイズ源を使用することである。同じ鍵番号によって種を蒔かれる同じ決定論的ノイズ源を、復号化処理において使用することができる。このような装置において、メモリ２１４に通常記憶される大きなデータセットの代わりに、後に復号化において使用するために鍵番号のみを記憶する必要がある。

代わりに、符号化中加算されたノイズ信号がゼロ平均値を有しておらず、コードワード長さＮがデコーダにとって既知である場合、万能復号化処理を行うことができる。この処理は、上述した手順と同様のエントロピ試験を使用するが、可能なコードワードを循環し、試験されているコードワードのビットにしたがって

、エントロピの減少が認められるまで、疑わしい信号のＮ番目の標本毎に小さいダミーノイズ値（例えば、予測される平均ノイズ値より小さい）を加算／減算する。しかしながら、このようなアプローチは、他の実施例より低い安全性しか示さない（例えば、野蛮な力によるクラッキングを受けやすい）ため、大部分の用途に対しては好適ではない。

多くの用途を、異なったコードワードを使用し、各々が同じノイズデータを使用する、入力信号のいくつかの異なるように符号化された変形を発生する図７に示した実施例によって取り扱うことができる。さらに特に、図７の実施例２４０は、ノイズ源２０６からのノイズを、第１コードワードによる入力信号の識別符号化中に記憶するノイズストア２４２を含む。（図７のノイズ源を、図の便宜上、リアルタイムエンコーダ２０２の外側に示す。）その後、入力信号の追加の検証符号化版を、前記ストアから記憶されたノイズデータを読み取り、Ｎ番目のコードワードを通じて交互に結合し、この信号を符号化することによって発生することができる。（２値逐次コードワードを図７に示すが、他の実施例においてコードワードの任意の配列を使用することができる。）このような装置によって、比例したサイズのロングタームノイズメモリを必要とすることなく、多くの数の

異なって符号化された信号を発生することができる。代わりに、一定量のノイズデータを記憶し、オリジナルを１回または１０００回符号化する。

（もし望むなら、いくつかの異なって符号化された出力信号を、順次ではなく同時に発生することができる。あるこのような実施は、各々が同じ入力信号および同じスケールノイズ信号によって駆動されるが、異なったコードワードによって駆動される複数の加算器／減算器を含む。この時各々は、異なって符号化された出力信号を発生する。）

同じオリジナルの多くの異なった符号化版を有する用途において、コードワードのすべてのビットを常に識別する必要はないことが認識されるだろう。例えば時々、用途は、疑わしい信号が属するコードのグループのみの検証を必要としてもよい。（例えば、コードワードの高次のビットは、同じソース作品のいくつかの異なった符号化版が発生された構造を示す低次のビットは、特定のコピーを示す。疑わしい信号が関係している構造を検証するために、構造を高次のビットの

みによって検証することができることから、低次のビットを調査する必要はない。）検証必要条件を、疑わしい信号におけるコードワードビットの部分集合を識別することによって満たすことができるならば、復号化処理を短縮することができる。

いくつかの用途を、あるときには異なったコードワードとともに、何回か積分作業中に、符号化処理を再開始することによって最適に取り扱うことができる。例として、ビデオテープ作品（例えば、テレビジョン番組）を考える。ビデオテープ作品の各々のフレームを、固有コード番号とともに検証符号化することができ、図８に示したのと同様の装置２４８によってリアルタイムで処理することができる。垂直帰線をシンク検出器２５０によって検出する度に、ノイズ源２０６をリセットし（例えば、丁度発生された配列を繰り返す）、検証コードを次の値に増加刷る。それによってビデオテープの各々のフレームは、固有に検証符号化される。代表的に、符号化信号を、長期間記憶するためにビデオテープに記憶する（レーザディスクを含む他の記憶媒体も使用することができる）。

符号化装置に戻ると、示した実施例における参照表２０４は、入力データ信号

の大振幅の標本は、小振幅入力標本ができるよりも高いレベルの符号化検証符号化を取り扱うことができるという事実を利用する。したがって例えば、０、１または２の１０進数値を有する入力データ標本を、１（またはゼロ）のスケール係数に対応させることができるが、２００を越える値を有する入力データ標本を、１５のスケール係数に対応させることができる。一般的に言って、スケール係数および入力標本値は、平方根関係によって対応する。すなわち、標本化入力信号の値における４つ折の増加は、これらに関係するスケール係数の値における２つ折の増加にほぼ対応する。

（ゼロのスケーリング係数に対する挿話的参照として、例えば、ソース信号が時間的または空間的に情報内容が無い場合に言及する。画像において、例えば、いくつかの隣接した０の標本値によって特徴付けられる領域を、フレームの真黒領域に対応させることができる。ゼロのスケーリング値を、著作権侵害される画像データが実際的にないことから、ここに充てることができ。）

符号化処理を続けると、当業者は、示した実施例における″レールエラー″に

対するポテンシャルを認識するであろう。例えば、入力信号が８ビット標本から成り、これらの標本が０から２５５（１０進数）の範囲全体に及んでいる場合、入力信号への／からのスケールノイズの加算／減算は、８ビットによっては表すことができない出力信号（例えば、－２または２５７）を発生するかもしれない。この状況を修正する多くの良く理解されている技術が存在し、これらのあるものは順行的であり、これらのあるものは反動的である。（これらの既知の技術は共通して、入力信号が０－４または２５１－２５５の範囲に標本を持たないようにし、それによってノイズ信号による変調を安全に行うか、他にレールエラーを発生する入力信号標本を検出し、適合するように変更する装置を含むかである。）

示した実施例は、コードワードを逐次に、一度に１ビットずつ進むことを記述するが、コードワードのビットをこの目的のために順次ではなく使用できることが理解できるであろう。実際に、コードワードのビットを、なんらかの予め決められたアルゴリズムに従って選択することができる。

入力信号の瞬間の値に基づくノイズ信号の動的なスケーリングは、多くの実施例において省略することができる最適化である。すなわち、参照表２０４および第１スケーラ２０８を完全に省略し、ディジタルノイズ源２０６からの信号を、加算器／減算器２１２に直接（または第２大域的スケーラ２１０を通して）供給することができる。

さらに、ゼロ平均ノイズ源の使用が示した実施例を簡単にすることが認識されるであろうが、本発明には必要ではない。他の平均値を有するノイズ信号を、容易に使用することができ、（もし必要なら）Ｄ．Ｃ．補正を、本システム以外で行うことができる。

ノイズ源２０６の使用も任意である。種々の他の信号源を、用途に応じて、制限（例えば、符号化検証信号が知覚できるようになるしきい値）に応じて使用することができる。多くの場合において、埋め込み検証信号のレベルは、検証信号がランダムな状況を有する必要がない、すなわちその性質にもかかわらず知覚できないほど十分に低い。しかしながら、埋め込み検証信号の知覚できないことのレベルに対して、最も大きな検証コード信号Ｓ／Ｎ比（この場合において、多少

不適切な言葉）を提供するため、擬似ランダム源２０６が通常望ましい。

検証符号化を、信号を（すなわち、米国著作権法の言葉において″実際の形式において一定の″）データとしての記憶された形式に減少した後で行う必要はないことが認識されるであろう。例えば、その演奏がしばしば不正に録音される人気音楽家の場合を考える。コンサートホールのスピーカを駆動する前にオーディオを検証符号化することによって、コンサートの認可されない録音を、個々の場所および時間まで追跡することができる。さらに、９１１非常呼び出しのような生のオーディオ源を、これらの後の認証を容易にするために、録音前に符号化することができる。

ブラックボックス実施例を独立型ユニットとして記述したが、多くの道具／器具中に構成要素として統合できることが認識されるであろう。その１つは、検証コードを走査した出力データ中に埋め込むことができるスキャナである。（これらのコードを、単にこのデータが個々のスキャナによって発生されたことを記念

するために取り扱うことができる）。他のものは、Ａｄｏｂｅ社、Ｍａｃｒｏｍｅｄｉａ社、Ｃｏｒｅｌ社、および同様の会社によって提供されている一般向けの描画／グラフィックス／アニメーション／ペイントプログラムのような創造的なソフトウエアにおけるものである。

最後に、リアルタイムエンコーダ２０２を個々のハードウエアの実装の参照とともに説明したが、種々の他の実装を代わりに使用できることが認識されるであろう。いくつかは、他のハードウエア形態を利用する。他のものは、説明した機能ブロックのいくつかまたはすべてに対してソフトウエアルーチンを使用する。（これらのソフトウエアルーチンを、８０ｘ８６ＰＣ互換コンピュータ、ＲＩＳＣベースのワークステーション、等のような多くの異なった一般的な目的のプログラム可能コンピュータにおいて実行することができる。

ノイズ、擬似ノイズ、および最適化ノイズの形式

これまで、本明細書は、画像または信号全体に渡って情報の１ビットを搬送するのに適切な搬送波信号の種類の多くの例の幾つかとして、ガウスノイズ、″ホワイトノイズ″、およひ用途器具から直接発生されたノイズを仮定した。ある目標を達成するために、ノイズの″設計″特性において、さらに順向的にすること

が可能である。ガウスまたは器具ノイズを使用する″設計″は、″絶対的″安全性のためにいくらか向けられている。本明細書のこの節では、検証情報の究極的な搬送波と考えることができるノイズ信号の設計のための、他の考察を調べる。

いくつかの用途に関して、搬送波信号（例えば、第１実施例におけるＮ番目の埋め込みコード信号、第２実施例におけるスケールノイズデータ）を、検証信号にこの信号の知覚可能性に関してより絶対的な信号強度を与えるために設計することが有利であるかもしれない。ある例は、以下のようなものである。真のガウスノイズは、値′０′が最も頻繁に生じ、次に１および－１が各々等しい確率だが′０′よりは低い確率で生じ、次に２および－２、等々である。明らかに、値０は、本発明において使用されるような情報を搬送しない。したがって、ある簡単な調節または設計は、埋め込みコード信号の発生においてゼロが発生するときはいつも、新たな処理が引き継ぎ、値を″ランダムに″１または－１のいずれか

に変換する。このような処理のヒストグラムは、０の値が空であり、１および－１の値が通常の０の値のヒストグラム値の半分だけ増加していることを除けば、ガウス／ポアソン型分布として現れる。

この場合において、検証信号エネルギは、通常、信号のすべての部分において現れる。交換のいくつかは、″決定論的成分″がノイズ信号の発生の一部であるコードの安全性の（大抵、無視できる）低下が存在することを含む。これを完全に無視できる理由は、我々が、１または－１をランダムに選択するコイン投げ形式の状況を依然として準備しているからである。他の交換は、設計されたノイズのこの形式が、知覚可能性の高いしきい値を有し、データストリームまたは画像の最下位ビットが題材の商業上価値に関してすでに無視できる、すなわち、最下位ビットが信号（またはすべての信号標本）から取り除かれた場合、誰もその差を識別できず、題材の価値が損害を受けない用途にのみ使用することができることである。上述した例におけるこのゼロ値の制限は、当業者の誰もが実現できるような信号搬送波のノイズ特性を″最適化″する多くの方法の１つである。我々は、これを、自然ノイズを予め決められた方法においてすべての意図および目的に対してノイズとして読み取られる信号に変換することができるという意味で″擬似ノイズ″とも呼ぶ。暗号化方法およびアルゴリズムが、完全にランダムとして知覚される信号を、容易に、そしてしばしば定義によって生成することもできる。したがって、″ノイズ″という言葉は、観察者または聴取者によって主観的に定義されるものと、数学的に定義されるものとの間で、異なった意味を有する。後者の違いは、数学的ノイズが、異なった安全性の性質を有し、追跡することができる簡単さか、このノイズの存在を″自動的に認識″することができる簡単さかを有する。

″万能″埋め込みコード

本明細書の大部分は、絶対的安全性のために、検証信号の情報のビットを搬送するノイズ様埋め込みコードを、各々すべての埋め込み信号に対して固有のものにすべきであるか、わずかに制限を少なくして、埋め込みコード信号を、例えばフィルムの１０００個の断片の１組に対して同じ埋め込みコードを使用するよう

に控えめに発生すべきであることを教えている。いずれにせよ、我々が″万能″埋め込みコード信号と呼ぶことができるものを使用することによって、この技術に関して新たな用途を大きく開発することができる他のアプローチが存在する。これらを使用することの経済性は、これらの万能コードの実際の低い信頼性（例えば、これらは、時間に頼った暗号復号化方法によって分析可能であり、したがって、可能的に妨げられるまたは置き換えられる）が、意図された使用を規定した場合の経済的利益と比較して経済的に無視できるようなものである。著作権侵害および非合法な使用は、単に、予測しうる″費用″および未徴収の収入源となり、すなわち全体の経済的分析における簡単なラインアイテムとなる。これの良い類似は、ケーブル産業とビデオ信号の波長を変えることとにおけるものである。一般に法律を甘受する市民である狡猾な技術的に熟練した個人が、全ての有料チャンネルをただにするためのケーブル接続ボックスにおいて、梯子をのぼり、数本のワイアをはじくことができることを誰もが知っていると思われる。ケーブル産業は、これを知っており、それを停止し、捕らえられたこれらを起訴する有効な方法を選択するが、この習慣に発する″失われた収入″は、いまだ普及しており、しかしシステム全体をスクランブルことによって得られる利益の割合としては、ほとんど無視できる。全体としてのスクランブル化システムは、″完全な安全性″の欠落にも係わらず、経済的に成功している。

同様なことが、この技術の用途に対して真実であり、ある程度の安全性を低下する価格に対して、大きな経済的機会をそれ自身に与える。この節は、最初に、万能コードによって何がもたらされるかを記述し、次に、これらのコードを用いることができるいくつかの興味深い使用に移る。

万能埋め込みコードを一般に、正確なコードの知識を配布することができるという概念に適用する。埋め込みコードを、（本明細書の他の部分において言及したように）訴訟がなされるまで決して接触されない秘密の金庫中に置かずに、代わりにその場で分析を行うことができる種々の場所に配布する。一般にこの配布は、安全性が制御された状況に依然として置かれており、ステップは、コードの認識が知ることを必要とするこれらに対して制限されることを意味する。著作権

を有する作品を自動的に検出しようとする方法は、コードを知ることを必要とする〃何か〃の人間でない例である。

万能コードの概念を実施する多くの方法が存在し、これらの各々が、何らかの所定の用途に関しては利点を有する。この技術を教える目的のために、我々は、これらのアプローチを３つのカテゴリー、すなわち、ライブラリを基礎とする万能コードと、決定論的公式を基礎とする万能コードと、予め規定された業界標準パターンを基礎とする万能コードとに分類する。おおざっぱなやり方は、第１のものは、後者の２つより安全性が高いが、後者の２つは、第１のものよりもより経済的に実現できるとする。

万能コード：１）万能コードのライブラリ

万能コードのライブラリの使用は、個々の埋め込みコード信号の制限された組のみが発生し、どのような所定の符号化材料もこの制限された〃万能コード〃の部分集合を使用することを除いて、本発明の技術を使用することを単に意味する。一例は、以下のものが適切である。写真印画紙製造業者は、固有検証コードとともに販売したい８×１０インチの印画紙のすべてを前露光することを望むことができる。彼らは、検証コード認識ソフトウエアを、彼らの大口顧客、サービス部、在庫代理店、および個々の写真家に販売し、その結果、すべてのこれらの人々が、かれらの題材が正確にマークされていることを照合できるだけでなく、彼らがまさに得ようとしている第三者の題材がこの技術によって著作権を取得して

いるとして確認された場合、決定することができるようにすることも望む。この後者の情報は、多くの他の利益のなかで、著作権所有者を確認し、訴訟を無効にするのを助ける。この計画を〃経済的に〃行うために、各々すべての印画紙に固有検証埋め込みコードを発生することは、情報とは独立に数テラバイトを発生し、これらのバイトを記憶する必要があり、これらのバイトに認識ソフトウエアがアクセスする必要がある。代わりに、彼らは、５０個の独立〃万能〃埋め込みコード信号のみの組から得た１６ビット検証コードを彼らの印画紙に埋め込むことを決める。これをどのように行うかについての詳細は、次の節におけるものであるが、かれらの認識ソフトウエアが、代表的に８×１０の印画紙上に広げられた

５０×１６の個々の埋め込みコードに対して（ディジタル圧縮を考慮して）１メガバイトから１０メガバイトの情報である、彼らのコードのライブラリにおける埋め込みコードの制限された組を含むことのみを必要とすることが、ここでの要点である。１６の代わりに５０を選ぶ理由は、安全性がわずかに増すためであり、すべての写真に対して同じ１６個の埋め込みコードにした場合、シリアル番号容量が２の１６乗に制限されるだけでなく、より少ない洗練された著作権侵害者が、これらのコードを解読し、ソフトウエアツールを使用してこれらを除去することができる。

この計画を実施するための多くの異なった方法が存在し、以下は好適な方法の１つである。企業経営の知識によって、埋め込みコード信号のための１インチ当たり３００画素の規準は、多くの用途に関して十分な解像度であると定義される。これは、復号埋め込みコード画像が、８×１０のシート上に極めて低いレベルにおいて露光すべき３０００×２４００画素を含むことを意味する。これは、７２０００００画素を与える。図５および６のブラックボックス手段において記述したような我々の交互配列符号化システムを使用すれば、各々の独立埋め込みコード信号は、１６分の７２０００００すなわち４５０ｋ程度の真の情報を搬送する画素、すなわち所定のラスタライン上のすべての１６番目の画素のみを含む。これらの値は、代表的に２から－２の範囲のディジタル数であり、符号３ビット数によって十分に記述される。このとき埋め込みコードの未加工の情報内容は、４５０ｋの３／８番目のバイト倍すなわち１７０キロバイト程度である。ディジタル圧縮によって、これをさらに減少することができる。これらの決定のすべては、近い将来になんらかの所定の用途によって規定される、本技術分野において既知の、標準工学最適化原理に属する。したがって、我々は、これらの５０個の独立埋め込みコードが数メガバイトに達することが分かる。これは、認識ソフトウエア内の万能コードの″ライブラリ″として配布するのに全く適度なレベルである。進歩した標準暗号化装置を、自称著作権侵害者が単に万能埋め込みコードをリバースエンジニアするために認識ソフトウエアを購入したことに１つが関係する場合、これらのコードの正確な特徴を隠すために使用することができる。認

識ソフトウエアは、本明細書において教えた認識技術を用いる前に、コードを簡単に復号化することができる。

認識ソフトウエアそれ自体は、種々の特徴を確かに有するが、行う中心的な仕事は、所定の画像中にある万能著作権コードが存在する場合、これを決定することである。鍵となる問題は、もしあるとすれば、合計５０個の万能コードのうちどの１６個が含まれているかということと、１６個が見つかった場合、これらのビット値は何かということとである。これらの問題の回答の決定における鍵変数は、整合と、回転と、拡大（スケール）と、範囲とである。助けとなるヒントが何もない大部分の一般的な場合において、すべての変数を、すべての相互結合に渡って独立して変化させるべきであり、５０個の万能コードの各々を、エントロピの減少が発生するかどうかを見つけるために、加算および減算によって検査すべきである。厳密に言えば、これは莫大な仕事であるが、疑わしいコピーと比較するオリジナル画像を有するような、または８×１０の印画紙に比例する画像のオリエンテーションおよび範囲を知ることのような、この仕事をはるかに簡単にする多くの有用なヒントが見つかり、簡単な整合技術によって、ある許容しうる程度に対する変数のすべてを決定することができる。このとき、エントロピにおけるなんらかの減少を見つけるために、５０個の万能コードを通して繰り返すことが単に必要である。１つを行った場合、他の１５個も行うべきである。５０個の万能コードの所定の順序を、ＩＤコードワードの最上位ビットから最下位ビットまでの順序に変換する設定をするために、プロトコルが必要である。したがって、我々が、万能コード番号″４″の存在を発見し、そのビット値が″０″であることを発見し、万能コード″１″から″３″が明確に存在しないことを発見した場合、我々のＮビットＩＤコード数の最上位ビットは″０″である。同様に、我々が、次の存在する最も低い万能コードが番号″７″であることを見つけ、それが″１″であることが分かった場合、我々の次の最上位ビットは″１″である。適切に行うと、このシステムは、印画紙在庫シリアル番号を、ある登録または印画紙自体の製造業者に登録している限り、著作権所有者まで明確に追跡することができる。すなわち、我々は、万能埋め込みコード４、７、１１、１２、１５

、１９、２１、２６、２７、２８、３４、３５、３７、３８、４０、および４８を使用し、埋め込みコード０１１０　０１０１　０１１１　０１００を有する印画紙が、カナダ在住の未知の野性動物写真家兼、氷河映画撮影技師であるＬｅｏｎａｒｄ　ｄｅ　Ｂｏｔｉｃｅｌｌｉの所有物であるという登録を調べる。彼が無税で登録した彼のフィルムおよび印画紙の在庫を、彼がこの在庫を購入したとき、馬鹿げた簡単なプロセスを行う〃郵便の必要がない〃製造会社が親切にも準備した封筒に入れる、数秒の仕事のため、我々はこれを知っている。Ｌｅｏｎａｒｄに著作権使用料を支払う必要がある誰かは、それが現れることをチェックし、確実に登録は、著作権使用料の支払いプロセスをそのサービスの一部として自動化する。

ある終点は、真に洗練された著作権侵害者と、違法の目的を持った他の者とが、種々の暗号化方法を実際に使用してこれらの万能コードを解読することができ、これらを販売し、コードを除去または歪ませるのを助けることができるソフトウエアおよびハードウエアツールを制作することである。しかしながら我々は、これらの方法を、本明細書の一部として教えない。とにかく、これは、万能コードの容易さとこれらが開く用途に支払う必要がある値段の１つである。

万能コード：２）決定論的公式を基礎とする万能コード

万能コードのライブラリは、万能コードを付けられている信号および画像の存在および身元を開く鍵としての数メガバイトの独立した一般的にランダムのデータを記憶および変換することを必要とする。代わりに、種々の決定論的公式を、ランダムデータ／画像フレームの発生に使用し、これらによって、これらのコードのすべてをメモリ内に記憶することと、〃５０個〃の万能コードの各々に質問

することとを回避することができる。決定論的公式は、所定の信号または画像中に存在することが一度知られているＩＤコードを決定する処理を高速化するのを助けることもできる。他方では、決定論的公式を、あまり洗練されていない著作権侵害者によって追跡することができる。一度追跡されると、これらを、インターネット上で１００個のニュースグループに掲示するように、より簡単に伝達することができる。これらは、追跡および公表をかまわない多くの用途には適切で

あり、独立万能埋め込みコードを発生する決定論的公式を、単にチケットとすることができる。

万能コード：３）″簡単な″万能コード

この分類は、はじめの２つを結合したものの一部であり、この技術の原理の真に大きな規模の実施に最大限向けたものである。この種類を使用する用途は、信頼できる安全性が、低費用で大きな規模の実施と、これが可能にする莫大な経済的利益とほどは重要ではない形式のものである。一例としての用途は、検証認識ユニットを適度に値付けされた（テレビジョンのような）家庭用オーディオおよびビデオ装置中に直接配置する。このような認識ユニットは、代表的に、オーディオおよび／またはビデオを監視してこれらの著作権検証コードを探し、そこから、記録可能性が与えられているか否か、または中央オーディオ／ビデオサービス提供者に伝送されるとともに毎月の送り状に配置される番組特定課金メータの増加のような判断に基づく簡単な決定を行う。さらに、バーおよび他の公共の場所における″ブラックボックス″が、著作権を持った題材を監視し（マイクロフォンによって聞き）、ＡＳＣＡＰ、ＢＭＩ、等によって使用される詳細な報告書を生成することができる。

簡単な万能コードの中心となる原理は、いくつかの基本的な業界標準の″ノイズ状″で継ぎ目のない繰り返しのパターンを、信号、画像、および画像列中に挿入し、安価な認識ユニットが、Ａ）著作権″フラグ″の存在を決定するか、Ｂ）Ａに追加して、より複雑な決定構成および動作を容易にすることができるようにすることである。

本発明のこの実施例を実現するために、独立埋め込みノイズ信号を発生する基本的な原理を、安価な認識信号処理ユニットに適応させると同時に、有効なランダム性およびホログラフィックの浸透の性質を維持するために、簡単にする必要がある。これらの簡単なコードの大規模産業への採用によって、コード自体は公有情報と隣接し（ケーブルスクランブリングボックスがほとんど事実上の公有であるように）、ブラックマーケット対策を開発するために確定された著作権侵害者に対してドアを開いたままであるが、この状況は、ケーブルビデオのスクラン

ブル化や、このような違法活動の客観的経済的分析とまったく類似している。

順向の著作権検出のこの一般的な領域における本出願人に既知のある先行技術は、オーディオ業界における多くの会社によって採用されたシリアルコピー管理システムである。本出願人の知っている限り、このシステムは、オーディオデータストリームの一部ではないが、それにもかかわらずオーディオストリームに挿入され、関連するオーディオデータを複製すべきか否かを示すことができる、非オーディオ″フラグ″信号を使用する。このシステムが有する１つの問題は、この追加の″フラグ″信号をサポートすることができる媒体および装置が制限されることである。他の欠陥は、フラグシステムが、より複雑な決定を行うのに使用できる身元情報を搬送しないことである。さらに他の困難は、アナログ信号の高品質なオーディオ標本化が、あるディジタルマスタの完全なディジタルコピーを任意に近く行えるようになる恐れがあり、この可能性を禁じる対策は、無いように思われる。

本発明の原理を、オーディオ用途、ビデオ、および上述した他のすべての用途における、これらのおよび他の問題を影響を与えることができる。簡単な万能コードの用途の一例は、以下のようなものである。ある１つの業界標準″１．００００００秒のノイズ″は、なんらかの所定のオーディオ信号の著作権符号の存在または不在を示す最も基本的なものとして規定される。図９は、業界標準ノイズ秒が時間領域４００および周波数領域４０２の双方においてどのように見えるかの一例である。定義によって、連続関数であり、標本化レートおよびビット量子化の何らかの組み合わせに適合する。規格化された振幅を有し、どのようなディジタル信号振幅にも任意に尺度合わせすることができる。この信号の信号レベルおよび最初のM番目の導関数は、２つの境界４０４において連続であり（図９ＣＣ）、その結果、繰り返す場合、信号における″不連続″は（波形として）目に

見えない、または、ハイエンドオーディオシステムによって演奏される場合、聞き取れない。１秒の選択は、この例において任意であり、この間隔の正確な長さを、可聴性、擬似ホワイトノイズ状態、継ぎ目のない繰り返し可能性、認識処理の容易さ、および著作権を付ける決定を行えることによる速度のような理由から

得る。この繰り返しノイズ信号の信号または画像への（再び、人間の知覚力以下のレベルにおける）挿入は、著作権題材の存在を示す。これは、本質的に１ビット検証コードであり、他の検証情報の埋め込みを、この節において後に論考する。この検証技術の使用を、ここで論考した低価格家庭向け器具を遙に越えて拡張することができ、スタジオにこの技術を使用することができ、監視局を設定し、実際に数１００チャネルの情報を同時に監視し、マークされた信号ストリームを探索し、さらに、課金ネットワークおよび印税追跡システムに適合する関連する身元コードを探索することができる。この基本的な標準化ノイズ署名を、継ぎ目無く何度も繰り返し、基本著作権検証をマークすべきオーディオ信号に加える。〝簡単〟という言葉の理由の一部は、以下のように理解される。明らかに著作権侵害者は、この業界標準信号について知るであろうが、削除または改ざんのようなこの知識から得られる彼らの違法な使用は、大規模な市場に対する全体的な技術の経済的な価値に比較して、経済的に非常に小さいものとなる。大部分のハイエンドオーディオに関して、この信号を、フルスケールから８０から１００ｄＢ低下またはさらに小さいものとし、各々の状況を、たとえ推薦されるものが確実に存在しても、それら自身のレベルに選択することができる。信号の振幅を、ノイズ署名が用いられているオーディオ信号レベルに従って変調することができる。すなわちこの振幅を、ドラムビートの場合、意味のある程度、しかし聞き取れるまたは不快になるほど劇的ではない程度に増加することができる。これらの程度は、記述すべき認識回路網を単に助ける。

このノイズ信号の存在の低価格な機器による認識を、種々の方法において行うことができる。あるものは、オーディオ信号出力の測定の簡単な原理に対する基本的な変形に基づいている。ソフトウエア認識プログラムを書くことができ、さらに洗練された数学的検出アルゴリズムを、より高い信頼性のある検証の検出を行うために用いることもできる。このような実施例において、著作権ノイズ署名

の検出は、オーディオ信号の時間平均された出力レベルと、ノイズ署名を減算した同じオーディオ信号の時間平均された出力レベルとの比較を含む。ノイズ信号を減算されたオーディオ信号が、変更されていないオーディオ信号より低い出力

レベルを有する場合、著作権署名が存在し、同じ意味で、ある状態フラグを設定する必要がある。この比較の実行において含まれる主な工学的に微妙なものは、オーディオの録音再生速度が不一致（例えば、ある機器は、正確に１秒間隔に関して０．５％″遅い″かもしれない）である処理と、何らかの所定のオーディオ中の一秒のノイズ署名の未知の位相の処理（基本的に、この″位相″は、０から１秒位までかもしれない）とを含む。上述した２つほど中心的なものではないがそれにもかかわらず説明すべき他の微妙なものは、認識回路が、オーディオ信号に元に埋め込まれたノイズ署名より大きい振幅のノイズ署名を減算すべきではないことである。幸運にも、これを、単に小さい振幅のノイズ信号のみを減算することによって実行することができ、出力レベルが低下した場合、これは、出力レベルにおける″谷に向かっている″しるしとなる。さらに他の関連する微妙なものは、出力レベルの変化が、全体の出力レベルに対して極めて小さく、計算を一般に適切なビット精度によって、例えば、時間平均された出力レベルにおいて１６－２０ビットオーディオにおいて、３２ビット値演算および集積によって行う必要があることである。

明らかに、低価格用途用のこの出力レベル比較処理回路を設計し組み立てることは、技術的最適化の仕事である。ある交換は、より低い価格と複雑さのために回路網に形成することができる″近道″に関する検証の実行における精度である。この認識回路網の機器内の配置の好適実施例は、その仕事用に注文生産した１つのプログラム可能集積回路によるものである。図１０は、あるこのような集積回路５０６を示す。ここで、オーディオ信号が、ディジタル信号として、またはＩＣ５００内でディジタル化すべきアナログ信号として５００中に入り、出力信号は、著作権ノイズ署名が見つかった場合にあるレベルに設定され、見つからなかった場合に他のレベルに設定されるフラグ５０２である。標準化ノイズ署名波形を、ＩＣ５０６内の読み出し専用メモリ５０４に記憶することも示す。オーディオ信号のＩＣ５０６への適用と、有効なフラグ５０２の出力との間には、認識

を行える前に、オーディオのある有限の位置を監視する必要があるため、僅かな時間遅延が存在する。この場合において、著作権ノイズ署名の存在または不在の

正確な決定を行うために十分な時間を有する場合、ＩＣが外界に知らせる″フラグ有効″出力信号が必要になるかもしれない。

図１０のＩＣ５０６の基本的な機能を実行するのに用いられる特定の設計および設計の哲学の広い範囲の変形例が存在する。オーディオ技術者およびディジタル信号処理技術者は、いくつかの基本的に異なった設計を生成することができる。あるこのような設計を図１１において、それ自体は、後に論考するような他の技術的最適化に属する処理５９９によって示す。図１１は、アナログ信号処理ネットワーク、ディジタル信号処理ネットワーク、またはソフトウエアプログラムのプログラミングステップのいずれかのフローチャートを示す。我々は、ある経路に沿った入力信号６００を、時間平均パワーメータ６０２に供給し、結果として得られるパワー出力それ自体を、信号P_{sig}として扱うことに気づく。右上に対して、我々は、６０４で通常速度の１２５％において読み取られ、したがってそのピッチが変化し、６０６で″ピッチ変化ノイズ信号″を示す、標準ノイズ署名５０４を見つける。次に、ステップ６０８において入力信号からこのピッチ変化ノイズ信号を減算し、この新たな信号を、６０２において示したのと同じ形式のここでは６１０で示す時間平均パワーメータに供給する。この操作の出力信号も、ここでは６１０で$P_{s\text{-}pcn}$と示す時間基準信号である。次にステップ６１２でパワー信号６１０からパワー信号６０２を減算し、パワー差信号P_{out}６１３を生じる。万能標準ノイズ署名が、入力オーディオ信号６００において実際に存在する場合、ケース２、６１８、が発生し、４秒間程度のビート信号が、出力信号６１３において現れ、図１２、６２２のようなステップによってこのビート信号を検出しなければならない。ケース１、６１４は、周期的なビートが見られない一様なノイズ信号である。ステップ６０４における１２５％を、ここでは任意に選択しており、技術的な理由が最適値を決定し、異なったビート信号周波数６１８を導く。この例における４秒の待機は、事実上一定期間であるが、特に少なくとも２つまたは３つのビートを検出したい場合、図１２は、図１１の基本設計を、隣から０．０５秒遅延されたオーディオの部分において各々一斉に動作する

２０個の並列回路によって１／２０秒程度遅延された入力信号の種々の遅延され

たバージョンにどのように繰り返し作用させるかの概要である。この方法において、ビート信号が、１／５秒程度毎に見られ、ビート検出回路の列を下る進行波のように見える。この進行ビート波の存在または不在は、検出フラグ５０２をトリガする。同時に、例えば、少なくとも２秒のオーディオが、フラグ有効信号５０８を設定する前に聞こえることを保証するオーディオ信号モニタが存在する。

オーディオの例を記述してきたが、ある繰り返し万能ノイズ信号または画像の同様の形式の定義を、多くの他の信号、画像、写真、およびすでに論考した物理的媒体に用いることができることは、当業者には明白であろう。

上述したケースは、情報の１ビット面のみを取り扱った。すなわち、ノイズ署名信号を、存在するか（１）、しないか（０）とした。多くの用途に関して、さらに複雑な判定か、または課金明細書におけるログ情報等に使用することができるシリアル番号情報をさらに検出することが好ましい。上述したのと同様の原理を用いるが、ここでは、図９に示すようなＮ個の独立ノイズ署名が、１つのこのような署名の代わりに存在する。代表的に、あるこのような署名は、これによって著作権マーキングが単に存在することを検出するマスタとし、これは一般に他のものより大きいパワーを有し、次に他のより小さいパワーの″検証″ノイズ署名をオーディオに埋め込む。認証回路は、一度主要なノイズ署名の存在を見つけると、他のＮ個のノイズ署名に進み、上述したものと同様のステップを用いる。ビート信号が検出される場合、これは１のビット値を示し、ビート信号が検出されない場合、これは０のビット値を示す。代表的にＮを３２とし、2^{32}個の検証コードを、本発明を使用する何らかの所定の産業に対して利用できるようにすることができる。

検証コードの長さが１である場合のこの技術の使用

本発明の原理を、１つの検証信号－－もし望むなら指紋－－の存在または不在のみを使用し、ある信号または画像が著作権を与えられていることの信頼性を与える場合において、明らかに適用することができる。業界標準ノイズ署名の上述した例は、ある適切な場合である。我々は、もはやコイン投げとの類似性の追加の信頼性を持たず、我々は、もはや追跡コード容量または基本シリアル番号容量

を持たないが、多くの用途は、これらの属性を必要としないであろうし、１つの指紋による追加の簡単さは、なんらかの事象におけるこれらの他の属性を補って余りある。

″壁紙″との類似性

″ホログラフィック″という言葉を、本明細書において、どのように検証コード番号を大部分完全な形態において符号化信号または画像全体に分布させるかを記述するのに使用してきた。これを、信号または画像の何らかの所定の断片は、完全な固有検証コード番号を含むという概念にも適用する。ホログラフィの物理的な実施の場合、この特性を失い始める前に、断片をどの位小さくできるかにおいて制限があり、ここでホログラフィック媒体の分解能制限は、ホログラフ自体に関する主要な要素である。図５の符号化装置を使用し、ゼロがランダムに１または－１に変化する上述した我々の″設計されたノイズ″をさらに使用する非改ざん配布信号の場合において、必要な断片の程度は、信号または画像ラスタラインにおいて単にＮ個の連続的な標本であり、ここでＮを、予め規定した我々の検証コード番号の長さであるとする。これは、情報の量であり、すなわち、ノイズおよび改ざんが作用する実際的な状況は、一般にこの簡単な数Ｎより１、２、または以上大きい桁の標本を必要とする。当業者は、これによって検証を行うことができる最も小さい断片の寸法における正確な統計の明確な定義に含まれる多くの変形が存在することを認識するであろう。

教授の目的のために、本出願人は、固有検証コード番号を、画像（または信号）を横切って″壁紙貼りした″というアナロジも使用する。すなわち、画像全体に何度も繰り返す。ＩＤコード番号のこの繰り返しを、図５のエンコーダの使用におけるように定期的にすることができ、またはそれ自身ランダムにすることができ、図６のＩＤコード２１６のビットは、通常の繰り返し方法において停止せず、各々の標本においてランダムに選択され、このランダムな選択は、出力信号２２８の値とともに記憶される。とにかく、ＩＤコードの情報キャリヤ、独立埋め込みコード信号は、画像または信号を横切って変化する。したがって、壁紙との類似性を要約すると、ＩＤコード自体を何度も繰り返すが、各々の繰り返しがつけるパターンは、一般に追跡できない鍵に従って、ランダムに変化する。

損失データ圧縮

上述したように、好適実施例の検証符号化は、損失データ圧縮およびその後の伸長とに耐えうる。このような圧縮は、特にディジタル化された娯楽番組（映画、等）のような状況における使用が益々増えると思われる。

本発明の好適実施例によって符号化されたデータは、出願人に既知のすべての形式の損失圧縮に耐えうるが、商業的に最も重要だと思われるものは、ＣＣＩＴＴ　Ｇ３、ＣＣＩＴＴ　Ｇ４、ＪＰＥＧ、ＭＰＥＧおよびＪＢＩＧ圧縮／伸長標準である。ＣＣＩＴＴ標準は、黒および白の文書の圧縮（例えば、ファクシミリおよび文書記憶）において広く使用されている。ＪＰＥＧは、静止画に最も広く使用されている。ＭＰＥＧは、動画に最も広く使用されている。ＪＢＩＧは、黒および白の像への使用に関して、ＣＣＩＴＴ標準の有望な後継者である。これらのような技術は、損失データ圧縮の分野において良く知られており、良い概略を、Pennebaker et al, JPEG, Still Image Data Compression Standard, Van Nostrand Reinhold, N.Y., 1993において見ることができる。

ステガノグラフィおよび、より複雑なメッセージまたは情報の伝送におけるこの技術の使用

本明細書は、信号全体に１つの検証コードの壁紙貼りと前記において呼んだものに集中する。これは、多くの用途に関して所望の特徴であると思われる。しかしながら、メッセージを通過させる、または適切な検証情報の極めて長い列を信号または画像中に埋め込むことが望ましい他の用途が存在する。多くのこれらの考えられる用途の１つは、所定の信号または画像がいくつかの異なったグループによって操作されることを意図され、画像の特定の領域が、各々のグループの適切な操作情報の検証および挿入に確保されている場合である。

これらの場合において、図６におけるコードワード２１６を、ある予め決められた方法において、信号または情報位置の関数として実際に変化させることができる。例えば、画像において、コードをディジタル画像の各々すべてのラスタラインに関して変更することができる。１６ビットコードワードを２１６とすることができるが、各々の走査ラインは新たなコードワードを有し、したがって４８０の走査ライン画像は９８０バイト（４８０×２バイト）メッセージを通過させ

ることができる。メッセージの受信者は、メモリ２１４に記憶されたノイズ信号にアクセスするか、使用されている符号化方法のノイズコードの万能コード構造を知る必要がある。本出願人の知る限り、これは、ステガノグラフィの成熟した領域の新規のアプローチである。

万能コードの前述の３つの用途のすべてにおいて、万能コードに加えて、短い（ひょっとすると８または１６ビット）秘密コードを追加することがしばしば望まれる。これは、洗練された著作権侵害者による万能コードの削除の可能性に対する他の僅かな量の安全性をユーザにもたらす。

結論

本発明の原理を用いることができる多くの数の異なった実施例を考慮して、詳細な実施例は、単に説明的なものであり、本発明の範囲を限定するものとして選択されるものではないことを認識されたい。むしろ、私は、私の発明として、以下の請求の範囲およびこれらの同等物の範囲と精神内とすることができる全てのこのような実施例を請求する。

APPENDIX A

```
#include "main.h"

#define XDIM 512L
#define XDIMR 512
#define YDIM 480L
#define BITS 8
#define RMS_VAL 5.0
#define NUM_NOISY 16
#define NUM_DEMOS 3
#define GRAD_THRESHOLD 10

struct char_buf {
    char filename[80];
    FILE *fp;
    fpos_t fpos;
    char buf[XDIMR];
};
struct uchar_buf {
    char filename[80];
    FILE *fp;
    fpos_t fpos;
    unsigned char buf[XDIMR];
};
struct int_buf {
    char filename[80];
    FILE *fp;
    fpos_t fpos;
    int buf[XDIMR];
};
struct cortex_s {
    char filename[80];
    FILE *fp;
    fpos_t fpos;
    unsigned char buf[XDIMR];
};

struct uchar_buf test_image;
struct char_buf snow_composite;
struct uchar_buf distributed_image;
struct uchar_buf temp_image;
struct int_buf temp_wordbuffer;
struct int_buf temp_wordbuffer2;
struct uchar_buf snow_images;
struct cortex_s cortex;

int demo=0;  /* which demo is being performed, see notes */

int our_code;  /* id value embedded onto image */
int found_code=0;  /* holder for found code*/


int waitvbb(void){
    while( (_inp(PORT_BASE)&8) );
    while( !(_inp(PORT_BASE)&8) );
    return(1);
}

int grabb(void){
    waitvbb();
    _outp(PORT_BASE+1,0);
    _outp(PORT_BASE,8);
    waitvbb();
    waitvbb();
    _outp(PORT_BASE,0x10);
    return(1);
```

```
}

int livee(void){
    _outp(PORT_BASE,0x00);
    return(1);
}


int live_video(void){
    livee();
    return(1);
}

int freeze_frame(void){
    grabb();
    return(1);
}

int grab_frame(struct uchar_buf *image){
    long i;

    grabb();
    fsetpos(image->fp, &image->fpos  );
    fsetpos(cortex.fp, &cortex.fpos  );
    for(i=0;i<YDIM;i++){
        fread(cortex.buf,sizeof(unsigned char),XDIMR,cortex.fp);
        fwrite(cortex.buf,sizeof(unsigned char),XDIMR,image->fp);
    }
    livee();
    return(1);
}

int wait_vertical_blanks(int number){
    long i;
    for(i=0;i<number;i++)waitvbb();
    return(1);
}


int clear_char_image(struct char_buf *charbuffer){
    long i,j;
    char *pchar;
    fpos_t tmp_fpos;

    fsetpos(charbuffer->fp, &charbuffer->fpos  );
    for(i=0;i<YDIM;i++){
        fgetpos(charbuffer->fp, &tmp_fpos  );
        pchar = charbuffer->buf;
        fread(charbuffer->buf,sizeof(char),XDIMR,charbuffer->fp);
        for(j=0;j<XDIM;j++) *(pchar++) = 0;
        fsetpos(charbuffer->fp, &tmp_fpos  );
        fwrite(charbuffer->buf,sizeof(char),XDIMR,charbuffer->fp);
    }
    return(1);
}

int display_uchar(struct uchar_buf *image,int stretch){
    unsigned char *pimage;
    unsigned char highest = 0;
    unsigned char lowest = 255;
    long i,j;
    double dtemp,scale,dlowest;
    fpos_t tmp_fpos;

    if(stretch){
        fsetpos(image->fp, &image->fpos  );
        fread(image->buf,sizeof(unsigned char),XDIMR,image->fp);
        fread(image->buf,sizeof(unsigned char),XDIMR,image->fp);
```

```
        for(i=2;i<(YDIM-2);i++){
            fread(image->buf,sizeof(unsigned char),XDIMR,image->fp);
            pimage = &image->buf[3];
            for(j=3;j<(XDIM-3);j++){
                if( *pimage > highest )highest = *pimage;
                if( *pimage < lowest )lowest = *pimage;
                pimage++;
            }
        }
        if(highest == lowest ){
            printf("something wrong in contrast stretch, zero
contrast");
            exit(1);
        }
        scale = 255.0 / ( (double)highest - (double)lowest );
        dlowest = (double)lowest;
        fsetpos(image->fp, &image->fpos  );
        for(i=0;i<YDIM;i++){
            fgetpos(image->fp, &tmp_fpos  );
            fread(image->buf,sizeof(unsigned char),XDIMR,image->fp);
            pimage = image->buf;
            for(j=0;j<XDIM;j++){
                dtemp = ((double)*pimage - dlowest)*scale;
                if(dtemp < 0.0)*(pimage++) = 0;
                else if(dtemp > 255.0)*(pimage++) = 255;
                else *(pimage++) = (unsigned char)dtemp;
            }
            fsetpos(image->fp, &tmp_fpos  );
            fwrite(image->buf,sizeof(unsigned
char),XDIMR,image->fp);
        }
    }


    fsetpos(image->fp, &image->fpos  );
    fsetpos(cortex.fp, &cortex.fpos  );
    for(i=0;i<YDIM;i++){
        fread(image->buf,sizeof(unsigned char),XDIMR,image->fp);
        fwrite(image->buf,sizeof(unsigned char),XDIMR,cortex.fp);
    }
    return(1);
}


int clear_int_image(struct int_buf *wordbuffer){
    long i,j;
    int *pword;
    fpos_t tmp_fpos;

    fsetpos(wordbuffer->fp, &wordbuffer->fpos  );
    for(i=0;i<YDIM;i++){
        fgetpos(wordbuffer->fp, &tmp_fpos  );
        pword = wordbuffer->buf;
        fread(wordbuffer->buf,sizeof(int),XDIMR,wordbuffer->fp);
        for(j=0;j<XDIM;j++) *(pword++) = 0;
        fsetpos(wordbuffer->fp, &tmp_fpos  );
        fwrite(wordbuffer->buf,sizeof(int),XDIMR,wordbuffer->fp);
    }
    return(1);
}

double find_mean_int(struct int_buf *wordbuffer){
    long i,j;
    int *pword;
    double mean=0.0;

    fsetpos(wordbuffer->fp, &wordbuffer->fpos  );
```

```
    for(i=0;i<YDIM;i++){
        pword = wordbuffer->buf;
        fread(wordbuffer->buf,sizeof(int),XDIMR,wordbuffer->fp);
        for(j=0;j<XDIM;j++) mean += (double) *(pword++);
    }
    mean /= ((double)XDIM * (double)YDIM);

    return(mean);
}

int add_uchar_to_int(struct uchar_buf *image,struct int_buf *word){
    unsigned char *pimage;
    int *pword;
    long i,j;
    fpos_t tmp_fpos;

    fsetpos(image->fp, &image->fpos  );
    fsetpos(word->fp, &word->fpos  );
    for(i=0;i<YDIM;i++){
        pword = word->buf;
        fgetpos(word->fp, &tmp_fpos  );
        fread(word->buf,sizeof(int),XDIMR,word->fp);
        pimage = image->buf;
        fread(image->buf,sizeof(unsigned char),XDIMR,image->fp);
        for(j=0;j<XDIM;j++) *(pword++) += (int)*(pimage++);
        fsetpos(word->fp, &tmp_fpos  );
        fwrite(word->buf,sizeof(int),XDIMR,word->fp);
    }
    return(1);
}


int add_char_to_uchar_creating_uchar(struct char_buf *cimage,
    struct uchar_buf *image,
    struct uchar_buf *out_image){
    unsigned char *pimage,*pout_image;
    char *pcimage;
    int temp;
    long i,j;

    fsetpos(image->fp, &image->fpos  );
    fsetpos(out_image->fp, &out_image->fpos  );
    fsetpos(cimage->fp, &cimage->fpos  );
    for(i=0;i<YDIM;i++){
        pcimage = cimage->buf;
        fread(cimage->buf,sizeof(char),XDIMR,cimage->fp);
        pimage = image->buf;
        fread(image->buf,sizeof(unsigned char),XDIMR,image->fp);
        pout_image = out_image->buf;
        for(j=0;j<XDIM;j++){
            temp = (int) *(pimage++) + (int) *(pcimage++);
            if(temp<0)temp = 0;
            else if(temp > 255)temp = 255;
            *(pout_image++) = (unsigned char)temp;
        }
        fwrite(out_image->buf,sizeof(unsigned
char),XDIMR,out_image->fp);
    }
    return(1);
}


int copy_int_to_int(struct int_buf *word2,struct int_buf *word){
    long i;

    fsetpos(word2->fp, &word2->fpos  );
```

```
    fsetpos(word->fp, &word->fpos  );
    for(i=0;i<YDIM;i++){
        fread(word->buf,sizeof(int),XDIMR,word->fp);
        fwrite(word->buf,sizeof(int),XDIMR,word2->fp);
    }
    return(1);
}


void get_snow_images(void){
    unsigned char *psnow,*ptemp;
    int number_snow_inputs;
    int temp,*pword,*pword2,bit;
    long i, j;
    double rms,dtemp;

    live_video(); /* device specific */

    printf("\n\nPlease point camera at a medium lit blank wall. ");
    printf("\nDefocus the lens a bit as well ");
    printf("\nIf possible, place the camera into its highest gain,
and ");
    printf("\nput the gamma to 1.0.");
    printf("  Ensure that the video is not saturated ");
    printf("\nPress any key when ready... ");

    while( !kbhit() );
    printf("\nNow finding difference frame rms value... ");

    /* subtract one image from another, find the rms difference */
    live_video();
    wait_vertical_blanks(2);
    grab_frame(&temp_image);
    live_video();
    wait_vertical_blanks(2);
    grab_frame(&distributed_image); /* use first image as buffer */

    rms = 0.0;
    fsetpos(temp_image.fp, &temp_image.fpos  );
    fsetpos(distributed_image.fp, &distributed_image.fpos  );
    for(i=0;i<YDIM;i++){
        ptemp = temp_image.buf;
        fread(temp_image.buf,sizeof(unsigned
char),XDIMR,temp_image.fp);
        psnow = distributed_image.buf;
        fread(distributed_image.buf,sizeof(unsigned
char),XDIMR,distributed_image.fp);
        for(j=0;j<XDIM;j++){
            temp = (int) *(psnow++)  -  (int) *(ptemp++);
            dtemp = (double)temp;
            dtemp *= dtemp;
            rms += dtemp;
        }
    }
    rms /= ( (double)XDIM * (double)YDIM );
    rms = sqrt(rms);
    printf("\n\nAn rms frame difference noise value of %lf was
found.",rms);
    printf("\nWe want at least %lf for good measure",RMS_VAL);
    /* we want rms to be at least RMS_VAL DN, so ...   */
    if(rms > RMS_VAL) number_snow_inputs = 1;
    else {
        dtemp = RMS_VAL / rms;
        dtemp *= dtemp;
        number_snow_inputs = 1 + (int)dtemp;
    }
    printf("\n%d images will achieve this noise
level",number_snow_inputs);
```

```
    /* now create each snowy image */

    printf("\nStarting to create snow pictures... \n");
    fsetpos(snow_images.fp, &snow_images.fpos  ); /* set on first
image*/
    for(bit = 0; bit < BITS; bit++){

        clear_int_image(&temp_wordbuffer);
        for(i=0;i<number_snow_inputs;i++){
            live_video();
            wait_vertical_blanks(2);
            grab_frame(&temp_image);
            add_uchar_to_int(&temp_image,&temp_wordbuffer);
        }

        clear_int_image(&temp_wordbuffer2);
        for(i=0;i<number_snow_inputs;i++){
            live_video();
            wait_vertical_blanks(2);
            grab_frame(&temp_image);
            add_uchar_to_int(&temp_image,&temp_wordbuffer2);
        }

        /* now load snow_images[bit] with the difference frame
biased by
        128 in an unsigned char form just to keep things clean */
        /* display it on cortex also */
        fsetpos(temp_wordbuffer2.fp, &temp_wordbuffer2.fpos  );
        fsetpos(temp_wordbuffer.fp, &temp_wordbuffer.fpos  );
        fsetpos(temp_image.fp, &temp_image.fpos  );
        for(i=0;i<YDIM;i++){
            pword = temp_wordbuffer.buf;
            fread(temp_wordbuffer.buf,sizeof(int),XDIMR,temp_wordbuf
fer.fp);
            pword2 = temp_wordbuffer2.buf;
            fread(temp_wordbuffer2.buf,sizeof(int),XDIMR,temp_wordbu
ffer2.fp);
            psnow = snow_images.buf;
            ptemp = temp_image.buf;
            for(j=0;j<XDIM;j++) {
                *(psnow++) = *(ptemp++) = (unsigned char)
(*(pword++) - *(pword2++) + 128);
            }
            fwrite(snow_images.buf,sizeof(unsigned
char),XDIMR,snow_images.fp);
            fwrite(temp_image.buf,sizeof(unsigned
char),XDIMR,temp_image.fp);
        }
        freeze_frame();
        display_uchar(&temp_image,0); /*1 signifies to stretch the
contrast*/
        printf("\rDone snowy %d   ",bit);
        wait_vertical_blanks(30);
    }

    return;
}




void loop_visual(void){
    unsigned char *psnow;
    char *pcomp;
    long i,j,count = 0;
    int ok=0,temp,bit,add_it;
```

```
    double scale = 1.0 / RMS_VAL;
    double dtemp,tmpscale;
    fpos_t tmp_fpos;

    /* initial rms of each snowy image should be around 5 to 10 DN.
    let's assume it is 5, and assume further that our acceptable
noise level of
    the full snowy composite should be approximately 1 DN, thus we
need to
    scale them down by approximately 5*BITS as a first guess, then
do the
    visual loop to zoom in on final acceptable value */


    printf("\n\n Now calculating initial guess at amplitude...
\n");
    while( !ok ){
        /* calculate snow_composite */
        /* clear composite */
        clear_char_image(&snow_composite);

        fsetpos(snow_images.fp, &snow_images.fpos  ); /* set on
first image*/
        for(bit=0;bit<BITS;bit++){
            j = 128 >> bit;
            if( our_code & j)add_it=1;
            else add_it=0;
            fsetpos(snow_composite.fp, &snow_composite.fpos  );
            for(i=0;i<YDIM;i++){
                psnow = snow_images.buf;
                fread(snow_images.buf,sizeof(unsigned
char),XDIMR,snow_images.fp);
                fgetpos(snow_composite.fp, &tmp_fpos  );

fread(snow_composite.buf,sizeof(char),XDIMR,snow_composite.fp);
                pcomp = snow_composite.buf;
                for(j=0;j<XDIM;j++) {
                    dtemp = ((double)*(psnow++) -128.0) * scale;
                    if(dtemp<0.0){
                        temp = -(int) fabs( -dtemp +0.5);
                    }
                    else {
                        temp = (int) fabs( dtemp +0.5);
                    }
                    if(temp > 127) {
                        temp = 127;
                    }
                    else if(temp < -128) {
                        temp = -128;
                    }
                    if(add_it){
                        *(pcomp++) += (char)temp;
                    }
                    else {
                        *(pcomp++) -= (char)temp;
                    }
                }
                fsetpos(snow_composite.fp, &tmp_fpos  );

fwrite(snow_composite.buf,sizeof(char),XDIMR,snow_composite.fp);
            }
            printf("\rDone snowy %d   ",bit);
        }

        /* add snow composite to test image to form dist image */
        add_char_to_uchar_creating_uchar(
            &snow_composite,
            &test_image,
```

```
            &distributed_image);

        /* display both and cue for putting scale down, up or ok */
        i=count = 0;
        printf("\n Depress any key to toggle, enter to move on...\n
");
        printf("\r Distributed Image...      ");
        display_uchar(&distributed_image,0);
        while( getch() != '\r' ){
            if( (count++) % 2){
                printf("\r Distributed Image...      ");
                display_uchar(&distributed_image,0);
            }
            else {
                printf("\r Original Image...      ");
                display_uchar(&test_image,0);
            }
        }
        printf("\nScale = %lf ",scale);
        printf("\nEnter new scale, or >1e6 for ok... ");
        scanf("%lf",&tmpscale);
        if(tmpscale > 1e6)ok=1;
        else scale = tmpscale;
    }
    /* distributed image now is ok; calculate actual snow_images
used and
    store in those arrays; */

    fsetpos(snow_images.fp, &snow_images.fpos  ); /* set on first
image*/
    printf("\nNow storing snow images as used... \n");
    for(bit=0;bit<BITS;bit++){
        for(i=0;i<YDIM;i++){
            psnow = snow_images.buf;
            fgetpos(snow_images.fp, &tmp_fpos  );
            fread(snow_images.buf,sizeof(unsigned
char),XDIMR,snow_images.fp);
            for(j=0;j<XDIM;j++) {
                dtemp = ((double)*psnow -128.0) * scale;
                if(dtemp<0.0){
                    temp = -(int) fabs( -dtemp +0.5);
                }
                else {
                    temp = (int) fabs( dtemp +0.5);
                }
                *(psnow++) = (unsigned char)(temp + 128);
            }
            fsetpos(snow_images.fp, &tmp_fpos  );
            fwrite(snow_images.buf,sizeof(unsigned
char),XDIMR,snow_images.fp);
        }
        printf("\rDone snowy %d   ",bit);
    }
    return;
}




double find_grad(struct int_buf *image,int load_buffer2){
    int buf1[XDIMR],buf2[XDIMR],buf3[XDIMR];
    int *pbuf1,*pbuf2,*pbuf3,*p2;
    double total=0.0,dtemp;
    long i, j;
    fpos_t tmp_pos;
```

```
        fsetpos(image->fp, &image->fpos );
        fgetpos(image->fp, &tmp_pos );

        fsetpos(temp_wordbuffer2.fp, &temp_wordbuffer2.fpos );

        for(i=1;i<(YDIM-1);i++){
            fsetpos(image->fp, &tmp_pos );
            fread(buf1,sizeof(int),XDIMR,image->fp);
            fgetpos(image->fp, &tmp_pos );
            fread(buf2,sizeof(int),XDIMR,image->fp);
            fread(buf3,sizeof(int),XDIMR,image->fp);
            pbuf1=buf1;
            pbuf2=buf2;
            pbuf3=buf3;
            p2 = temp_wordbuffer2.buf;

        if(load_buffer2){
            for(j=1;j<(XDIM-1);j++){
                dtemp = (double)*(pbuf1++);
                dtemp += (double)*(pbuf1++);
                dtemp += (double)*(pbuf1--);
                dtemp += (double)*(pbuf2++);
                dtemp -= (8.0 * (double) *(pbuf2++));
                dtemp += (double)*(pbuf2--);
                dtemp += (double)*(pbuf3++);
                dtemp += (double)*(pbuf3++);
                dtemp += (double)*(pbuf3--);
                *p2 = (int)dtemp;
                if( *p2 > GRAD_THRESHOLD ){
                    *(p2++) -= GRAD_THRESHOLD;
                }
                else if( *p2 < -GRAD_THRESHOLD ){
                    *(p2++) += GRAD_THRESHOLD;
                }
                else {
                    *(p2++) = 0;
                }
            }

fwrite(temp_wordbuffer2.buf,sizeof(int),XDIMR,temp_wordbuffer2.fp);
        }
        else {

fread(temp_wordbuffer2.buf,sizeof(int),XDIMR,temp_wordbuffer2.fp);
            for(j=1;j<(XDIM-1);j++){
                dtemp = (double)*(pbuf1++);
                dtemp += (double)*(pbuf1++);
                dtemp += (double)*(pbuf1--);
                dtemp += (double)*(pbuf2++);
                dtemp -= (8.0 * (double) *(pbuf2++));
                dtemp += (double)*(pbuf2--);
                dtemp += (double)*(pbuf3++);
                dtemp += (double)*(pbuf3++);
                dtemp += (double)*(pbuf3--);

                dtemp -= (double) *(p2++);

                dtemp *= dtemp;
                total += dtemp;
            }
        }
        }

        return(total);
}
```

```
void search_1(struct uchar_buf *suspect){
    unsigned char *psuspect,*psnow;
    int bit,*pword,temp;
    long i,j;
    double add_metric,subtract_metric;
    fpos_t tmp_fpos;

    /* this algorithm is conceptually the simplest.  The idea is to
step
    through each bit at a time and merely see if adding or
subtracting the
    individual snowy picture minimizes some 'contrast' metric.
    This should be the most crude and inefficient, no where to go
but
    better */

    fsetpos(snow_images.fp, &snow_images.fpos  );
    temp=256;
    clear_int_image(&temp_wordbuffer);
    add_uchar_to_int(suspect,&temp_wordbuffer);
    find_grad(&temp_wordbuffer,1); /* 1 means load temp_wordbuffer2
*/
    for(bit=0;bit<BITS;bit++){
        /* add first */
        fgetpos(snow_images.fp, &tmp_fpos  );
        fsetpos(suspect->fp, &suspect->fpos  );
        fsetpos(temp_wordbuffer.fp, &temp_wordbuffer.fpos  );
        for(i=0;i<YDIM;i++){
            pword = temp_wordbuffer.buf;
            psuspect = suspect->buf;
            psnow = snow_images.buf;
            fread(suspect->buf,sizeof(unsigned
char),XDIMR,suspect->fp);
            fread(snow_images.buf,sizeof(unsigned
char),XDIMR,snow_images.fp);
            for(j=0;j<XDIM;j++){
                *(pword++)=(int)*(psuspect++)+(int)*(psnow++)-128;
            }
            fwrite(temp_wordbuffer.buf,sizeof(int),XDIMR,temp_wordbu
ffer.fp);
        }
        add_metric = find_grad(&temp_wordbuffer,0);

        /* then subtract */
        fsetpos(snow_images.fp, &tmp_fpos  );
        fsetpos(suspect->fp, &suspect->fpos  );
        fsetpos(temp_wordbuffer.fp, &temp_wordbuffer.fpos  );
        for(i=0;i<YDIM;i++){
            pword = temp_wordbuffer.buf;
            psuspect = suspect->buf;
            psnow = snow_images.buf;
            fread(suspect->buf,sizeof(unsigned
char),XDIMR,suspect->fp);
            fread(snow_images.buf,sizeof(unsigned
char),XDIMR,snow_images.fp);
            for(j=0;j<XDIM;j++){
                *(pword++)=(int)*(psuspect++)-(int)*(psnow++)+128;
            }
            fwrite(temp_wordbuffer.buf,sizeof(int),XDIMR,temp_wordbu
ffer.fp);
        }
        subtract_metric = find_grad(&temp_wordbuffer,0);

        printf("\nbit place %d: add=%le ,
sub=%le",bit,add_metric,subtract_metric);
        temp/=2;
        if(add_metric < subtract_metric){
            printf("  bit value = 0");
```

```
            }
            else {
                printf("  bit value = 1");
                found_code += temp;
            }
        }
        printf("\n\nYour magic number was %d",found_code);
        return;
}


void search_2(unsigned char *suspect){

    if(suspect);

    return;
}



void loop_simulation(void){
    unsigned char *ptemp,*pdist;
    int *pword,int_mean,ok=0,temp;
    long i,j;
    double mean,scale;

    /* grab a noisy image into one of the temp buffers */
    printf("\ngrabbing noisy frame...\n");
    clear_int_image(&temp_wordbuffer);
    for(i=0;i<NUM_NOISY;i++){
        live_video();
        wait_vertical_blanks(2);
        grab_frame(&temp_image);
        add_uchar_to_int(&temp_image,&temp_wordbuffer);
        j=(long)NUM_NOISY;
        printf("\r%ld of %ld     ",i+1,j);
    }

    /* find mean value of temp_wordbuffer */
    mean = find_mean_int(&temp_wordbuffer);
    int_mean = (int)mean;

    /* now we will add scaled version of this 'corruption' to our
distributed
    image */
    scale = 1.0;
    while( !ok ){
        /* add noise to dist image storing in temp_image */
        fsetpos(distributed_image.fp, &distributed_image.fpos  );
        fsetpos(temp_wordbuffer.fp, &temp_wordbuffer.fpos  );
        fsetpos(temp_image.fp, &temp_image.fpos  );
        for(i=0;i<YDIM;i++){
            pdist = distributed_image.buf;
            pword = temp_wordbuffer.buf;
            ptemp = temp_image.buf;
            fread(distributed_image.buf,sizeof(unsigned
char),XDIMR,distributed_image.fp);
            fread(temp_wordbuffer.buf,sizeof(int),XDIMR,temp_wordbuf
fer.fp);
            for(j=0;j<XDIM;j++){
                temp = (int) *(pdist++) + *(pword++) - int_mean;
                if(temp<0)temp = 0;
                else if(temp > 255)temp = 255;
                *(ptemp++) = (unsigned char)temp;
            }
```

```
            fwrite(temp_image.buf,sizeof(unsigned
char),XDIMR,temp_image.fp);
        }


        /* display the dist image and the corrupted image */
        display_uchar(&temp_image,0);

        /* apply new 'corrupted' image to search algorithm 1 for id
value */
        search_1(&temp_image);

        /* apply new 'corrupted' image to search algorithm 2 for id
value */
        /*
        search_2(temp_image);
        */

        /* prompt for upping noise content or ok */
        ok = 1;
    }

    return;
}



int initialize_everything(void){
    long i,j;
    unsigned char *pucbuf;
    char *pcbuf;
    int *pibuf;

    /* initialize cortex */
    strcpy(cortex.filename,"f:image");
    if((cortex.fp=fopen(cortex.filename,"rb"))==NULL){
        system("v f g");
    }
    else fclose(cortex.fp);
    if( (_inp(PORT_BASE) == 0xFF) ){
        printf("oops ");
        exit(0);
    }

    /* open cortex for read and write */
    if((cortex.fp=fopen(cortex.filename,"rb+"))==NULL){
        printf(" No good on open file joe ");
        exit(0);
    }
    fgetpos(cortex.fp, &cortex.fpos  );

/* test_image;  original image */
    strcpy(test_image.filename,"e:tst_img");
    if((test_image.fp=fopen(test_image.filename,"wb"))==NULL){
        printf(" No good on open file joe ");
        exit(0);
    }
    pucbuf = test_image.buf;
    for(i=0;i<XDIM;i++)*(pucbuf++)=0;
    for(i=0;i<YDIM;i++){
        fwrite(test_image.buf,sizeof(unsigned
char),XDIMR,test_image.fp);
    }
    fclose(test_image.fp);
    if((test_image.fp=fopen(test_image.filename,"rb+"))==NULL){
        printf(" No good on open file joe ");
        exit(0);
```

```
        }
        fgetpos(test_image.fp, &test_image.fpos  );

    /* snow_composite;  ultimate image added to original image */
        strcpy(snow_composite.filename,"e:snw_cmp");

    if((snow_composite.fp=fopen(snow_composite.filename,"wb"))==NULL){
            printf(" No good on open file joe ");
            exit(0);
        }
        pcbuf = snow_composite.buf;
        for(i=0;i<XDIM;i++)*(pcbuf++)=0;
        for(i=0;i<YDIM;i++){

    fwrite(snow_composite.buf,sizeof(char),XDIMR,snow_composite.fp);
        }
        fclose(snow_composite.fp);

    if((snow_composite.fp=fopen(snow_composite.filename,"rb+"))==NULL){
            printf(" No good on open file joe ");
            exit(0);
        }
        fgetpos(snow_composite.fp, &snow_composite.fpos  );

    /* distributed_image;   test_img plus snow_composite */
        strcpy(distributed_image.filename,"e:dst_img");

    if((distributed_image.fp=fopen(distributed_image.filename,"wb"))==NU
    LL){
            printf(" No good on open file joe ");
            exit(0);
        }
        pucbuf = distributed_image.buf;
        for(i=0;i<XDIM;i++)*(pucbuf++)=0;
        for(i=0;i<YDIM;i++){
            fwrite(distributed_image.buf,sizeof(unsigned
    char),XDIMR,distributed_image.fp);
        }
        fclose(distributed_image.fp);

    if((distributed_image.fp=fopen(distributed_image.filename,"rb+"))==N
    ULL){
            printf(" No good on open file joe ");
            exit(0);
        }
        fgetpos(distributed_image.fp, &distributed_image.fpos  );


    /* temp_image;   buffer if needed */
        strcpy(temp_image.filename,"e:temp_img");
        if((temp_image.fp=fopen(temp_image.filename,"wb"))==NULL){
            printf(" No good on open file joe ");
            exit(0);
        }
        pucbuf = temp_image.buf;
        for(i=0;i<XDIM;i++)*(pucbuf++)=0;
        for(i=0;i<YDIM;i++){
            fwrite(temp_image.buf,sizeof(unsigned
    char),XDIMR,temp_image.fp);
        }
        fclose(temp_image.fp);
        if((temp_image.fp=fopen(temp_image.filename,"rb+"))==NULL){
            printf(" No good on open file joe ");
            exit(0);
        }
        fgetpos(temp_image.fp, &temp_image.fpos  );

    /*  temp_wordbuffer;   16 bit image buffer for averaging */
```

```
    strcpy(temp_wordbuffer.filename,"e:temp_wrd");

if((temp_wordbuffer.fp=fopen(temp_wordbuffer.filename,"wb"))==NULL){
        printf(" No good on open file joe ");
        exit(0);
    }
    pibuf = temp_wordbuffer.buf;
    for(i=0;i<XDIM;i++)*(pibuf++)=0;
    for(i=0;i<YDIM;i++){

fwrite(temp_wordbuffer.buf,sizeof(int),XDIMR,temp_wordbuffer.fp);
    }
    fclose(temp_wordbuffer.fp);

if((temp_wordbuffer.fp=fopen(temp_wordbuffer.filename,"rb+"))==NULL)
{
        printf(" No good on open file joe ");
        exit(0);
    }
    fgetpos(temp_wordbuffer.fp, &temp_wordbuffer.fpos  );

/* temp_wordbuffer2;  /* 16 bit image buffer for averaging */
    strcpy(temp_wordbuffer2.filename,"e:tmp_wrd2");

if((temp_wordbuffer2.fp=fopen(temp_wordbuffer2.filename,"wb"))==NULL
){
        printf(" No good on open file joe ");
        exit(0);
    }
    pibuf = temp_wordbuffer2.buf;
    for(i=0;i<XDIM;i++)*(pibuf++)=0;
    for(i=0;i<YDIM;i++){

fwrite(temp_wordbuffer2.buf,sizeof(int),XDIMR,temp_wordbuffer2.fp);
    }
    fclose(temp_wordbuffer2.fp);

if((temp_wordbuffer2.fp=fopen(temp_wordbuffer2.filename,"rb+"))==NUL
L){
        printf(" No good on open file joe ");
        exit(0);
    }
    fgetpos(temp_wordbuffer2.fp, &temp_wordbuffer2.fpos  );

/* snow_images;  BITS number of constituent snowy pictures */
    strcpy(snow_images.filename,"snw_imgs");
    if((snow_images.fp=fopen(snow_images.filename,"wb"))==NULL){
        printf(" No good on open file joe ");
        exit(0);
    }
    pucbuf = snow_images.buf;
    for(i=0;i<XDIM;i++)*(pucbuf++)=0;
    for(j=0;j<BITS;j++){
    for(i=0;i<YDIM;i++){
        fwrite(snow_images.buf,sizeof(unsigned
char),XDIMR,snow_images.fp);
    }
    }
    fclose(snow_images.fp);
    if((snow_images.fp=fopen(snow_images.filename,"rb+"))==NULL){
        printf(" No good on open file joe ");
        exit(0);
    }
    fgetpos(snow_images.fp, &snow_images.fpos  );

    return(1);
}
```

```
int close_everything(void){

    fclose(test_image.fp);
    fclose(snow_composite.fp);
    fclose(distributed_image.fp);
    fclose(temp_image.fp);
    fclose(temp_wordbuffer.fp);
    fclose(temp_wordbuffer2.fp);
    fclose(snow_images.fp);

    return(1);
}


main(){
    int i,j;

    printf("\nInitializing...\n\n");
    initialize_everything();  /* device specific and global mallocs
*/

    live_video();

    /* prompt for which of the three demos to perform */
    while( demo < 1 || demo > NUM_DEMOS){
        printf("Which demo do you want to run?\n\n");
        printf("1: Digital Imagery and Very High End Photography
Simulation\n");
        printf("2: Pre-exposed Print Paper and other Dupping\n");
        printf("3: Pre-exposed Original Film (i.e. In-Camera)\n");
        printf("\nEnter number and return:  ");
        scanf("%d",&demo);
        if(demo < 1 || demo > NUM_DEMOS){
            printf("\n eh eh  ");
        }
    }

    /* acquire test image */
    printf("\nPress any key after your test scene is ready... ");
    getch();
    grab_frame(&test_image);  /*grab_frame takes care of device
specific stuff*/

    /* prompt for id number, 0 through 255 */
    printf("\nEnter any number between 0 and 255.\n");
    printf("This will be the unique magic code placed into the
image:  ");
    scanf("%d",&our_code);
    while(our_code<1 || our_code>256){
        printf(" Between 0 and 255 please  ");
        scanf("%d",&our_code);
    }

    /* feed back the binary code which will be embedded in the image
*/
    printf("\nThe binary sequence ");
    for(i=0;i<BITS;i++){
        j = 128 >> i;
        if( our_code & j)printf("1");
        else printf("0");
    }
    printf(" (%d) will be embedded on the image\n",our_code);

    /* now generate the individual snow images  */
    get_snow_images();
```

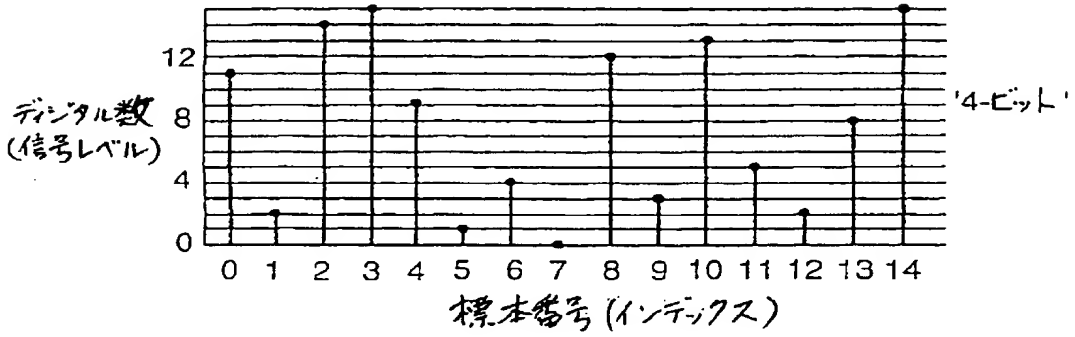
```
    loop_visual();  /* this gives visual feedback on 'tolerable'
noise level */

    printf("\nWe're now to the simulated suspect... \n");
    loop_simulation();

    close_everything();
    return(0);
}
```

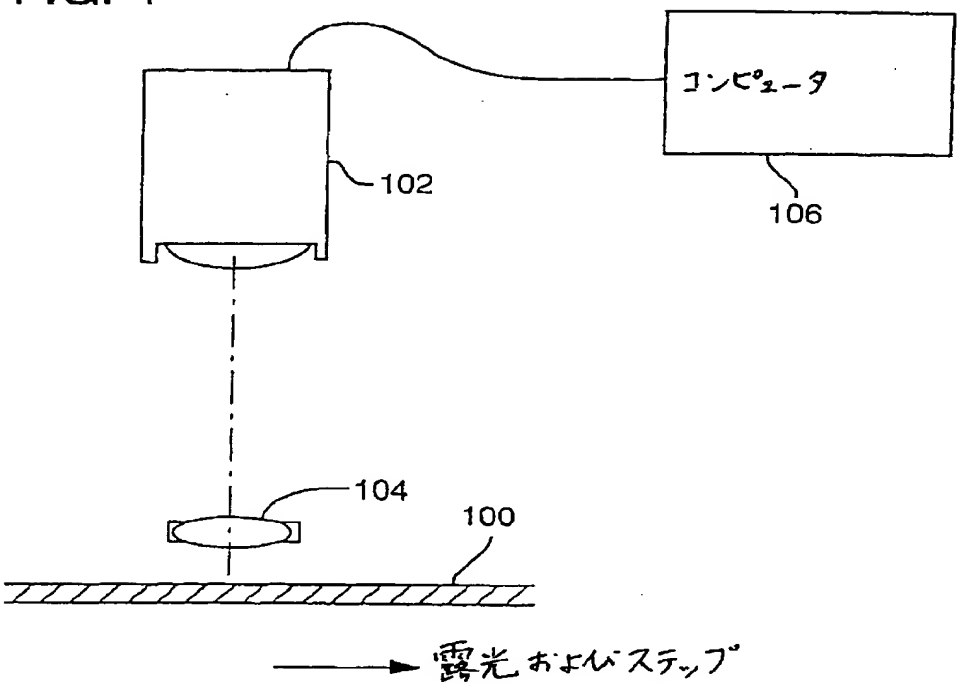
【図１】

FIG. 1



【図４】

FIG. 4



【図２】

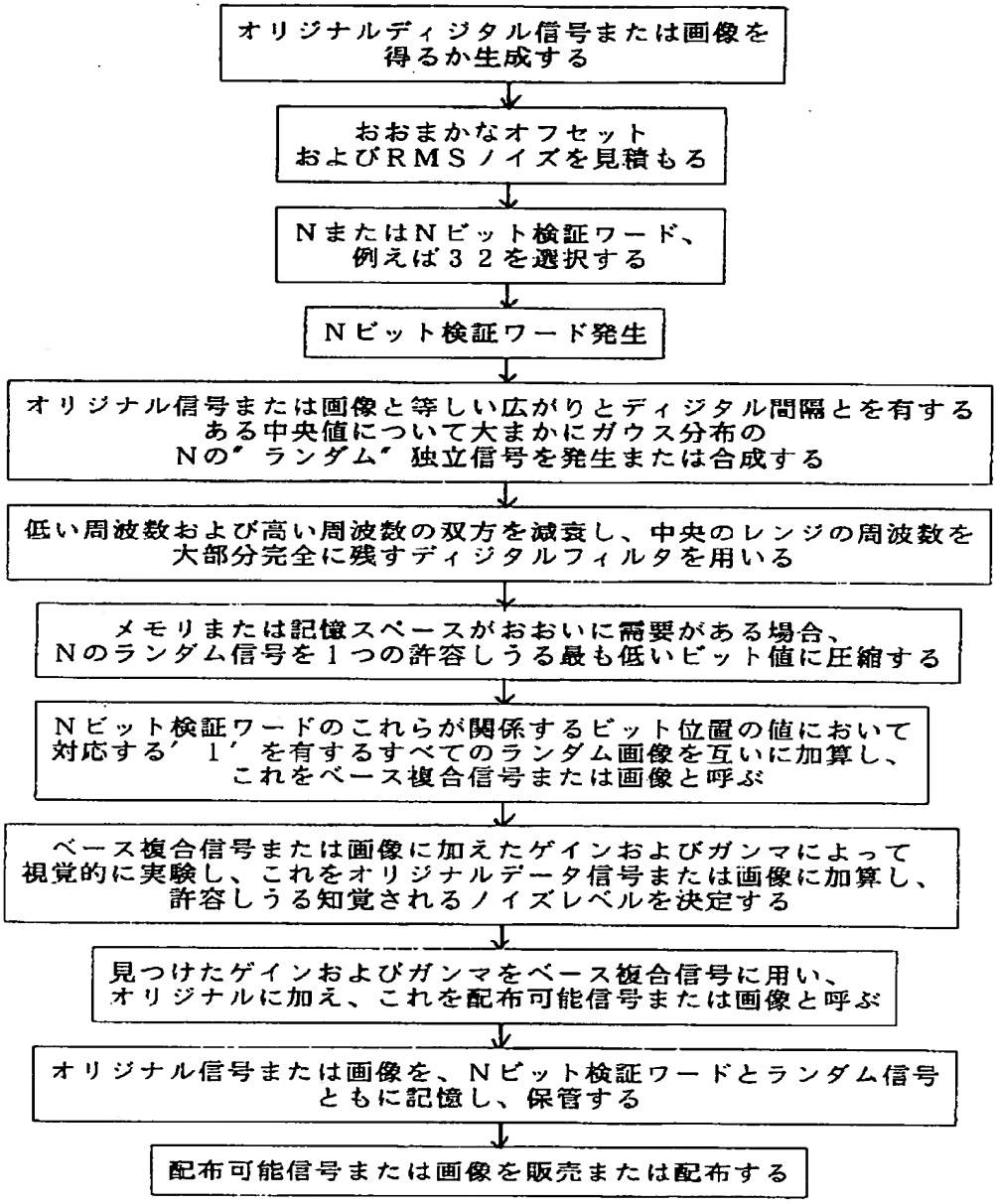

オリジナルディジタル信号または画像を得るか生成する
おおまかなオフセットおよびＲＭＳノイズを見積もる
ＮまたはＮビット検証ワード、例えば３２を選択する
Ｎビット検証ワード発生
オリジナル信号または画像と等しい広がりとディジタル間隔とを有するある中央値について大まかにガウス分布のＮの〝ランダム〟独立信号を発生または合成する
低い周波数および高い周波数の双方を減衰し、中央のレンジの周波数を大部分完全に残すディジタルフィルタを用いる
メモリまたは記憶スペースがおおいに需要がある場合、Ｎのランダム信号を１つの許容しうる最も低いビット値に圧縮する
Ｎビット検証ワードのこれらが関係するビット位置の値において対応する′１′を有するすべてのランダム画像を互いに加算し、これをベース複合信号または画像と呼ぶ
ベース複合信号または画像に加えたゲインおよびガンマによって視覚的に実験し、これをオリジナルデータ信号または画像に加算し、許容しうる知覚されるノイズレベルを決定する
見つけたゲインおよびガンマをベース複合信号に用い、オリジナルに加え、これを配布可能信号または画像と呼ぶ
オリジナル信号または画像を、Ｎビット検証ワードとランダム信号ともに記憶し、保管する
配布可能信号または画像を販売または配布する

FIG. 2

【図３】

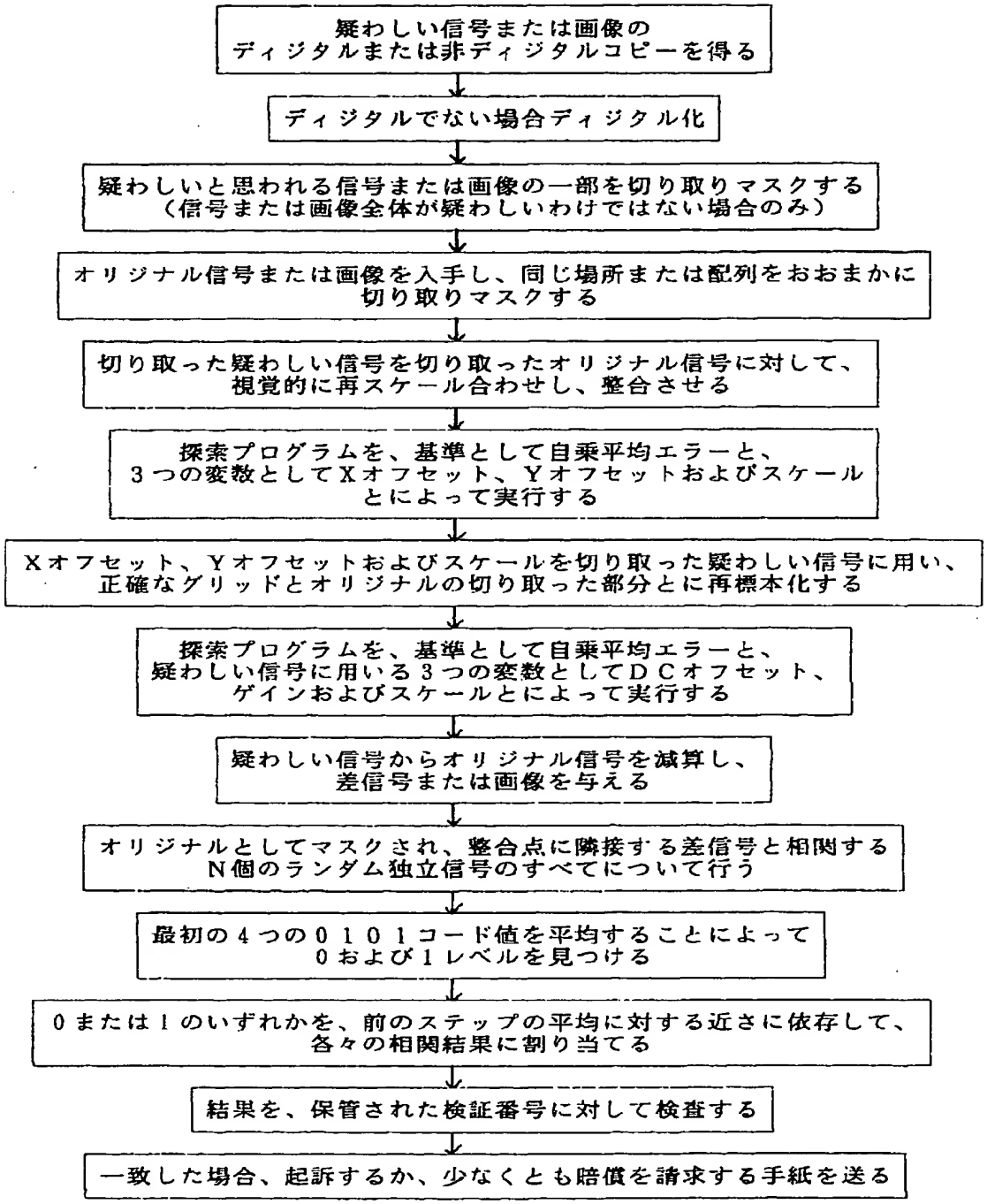
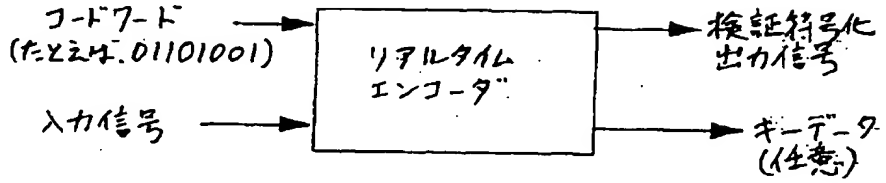

疑わしい信号または画像の
ディジタルまたは非ディジタルコピーを得る
ディジタルでない場合ディジタル化
疑わしいと思われる信号または画像の一部を切り取りマスクする
（信号または画像全体が疑わしいわけではない場合のみ）
オリジナル信号または画像を入手し、同じ場所または配列をおおまかに
切り取りマスクする
切り取った疑わしい信号を切り取ったオリジナル信号に対して、
視覚的に再スケール合わせし、整合させる
探索プログラムを、基準として自乗平均エラーと、
３つの変数としてＸオフセット、Ｙオフセットおよびスケール
とによって実行する
Ｘオフセット、Ｙオフセットおよびスケールを切り取った疑わしい信号に用い、
正確なグリッドとオリジナルの切り取った部分とに再標本化する
探索プログラムを、基準として自乗平均エラーと、
疑わしい信号に用いる３つの変数としてＤＣオフセット、
ゲインおよびスケールとによって実行する
疑わしい信号からオリジナル信号を減算し、
差信号または画像を与える
オリジナルとしてマスクされ、整合点に隣接する差信号と相関する
Ｎ個のランダム独立信号のすべてについて行う
最初の４つの０１０１コード値を平均することによって
０および１レベルを見つける
０または１のいずれかを、前のステップの平均に対する近さに依存して、
各々の相関結果に割り当てる
結果を、保管された検証番号に対して検査する
一致した場合、起訴するか、少なくとも賠償を請求する手紙を送る

FIG. 3

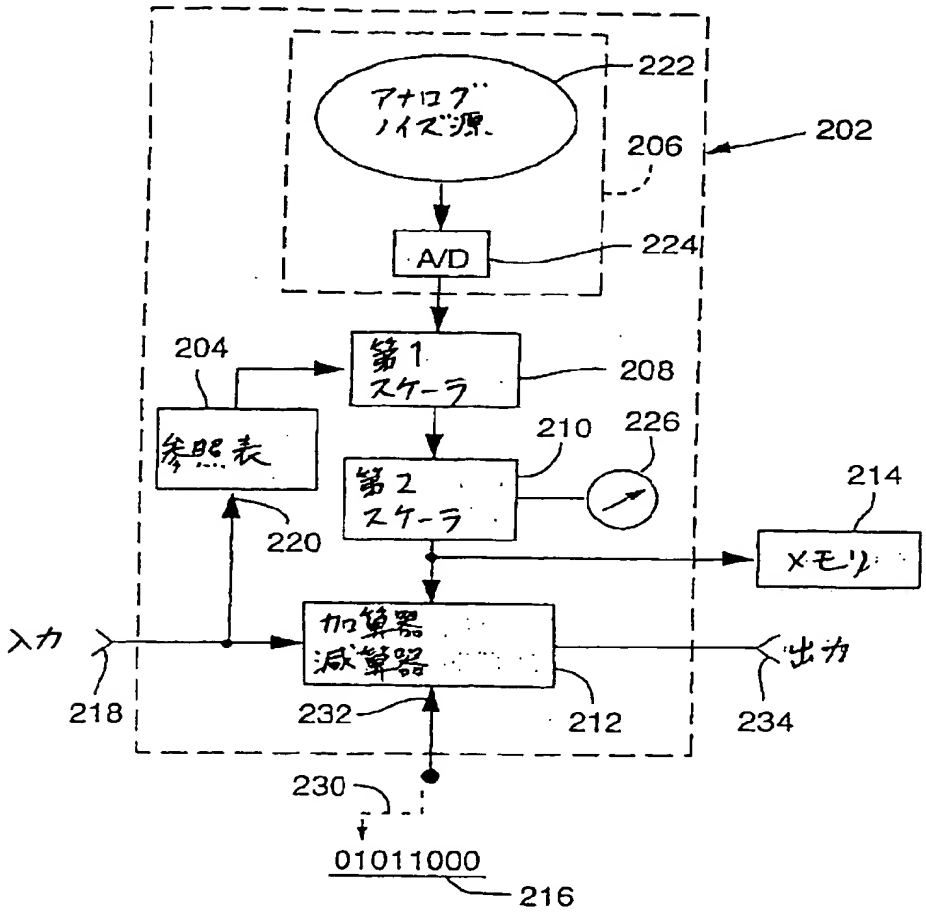
【図５】

FIG. 5



【図６】

FIG. 6



【図７】

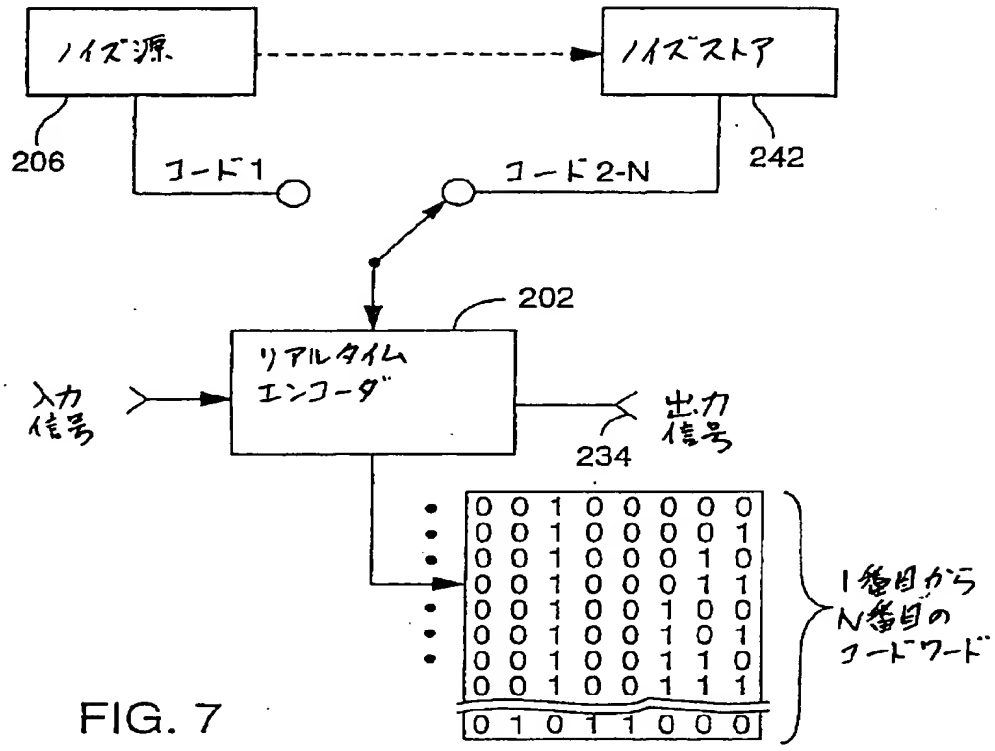
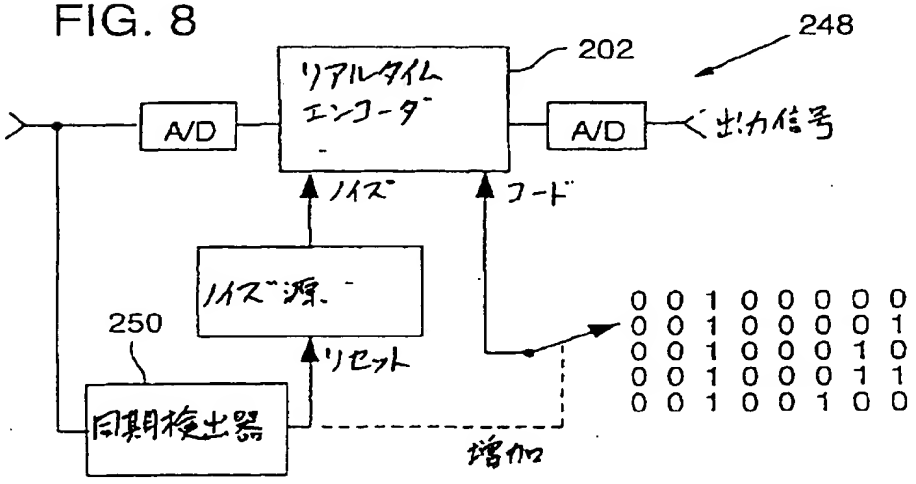


FIG. 7

【図８】

FIG. 8



【図９】

FIG. 9A

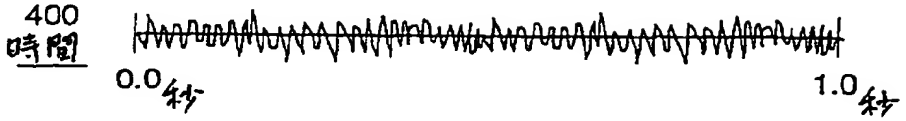


FIG. 9B

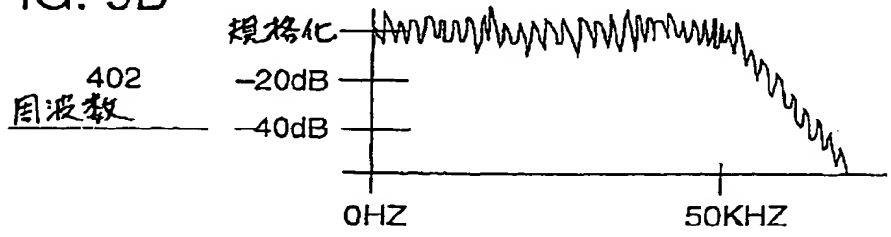
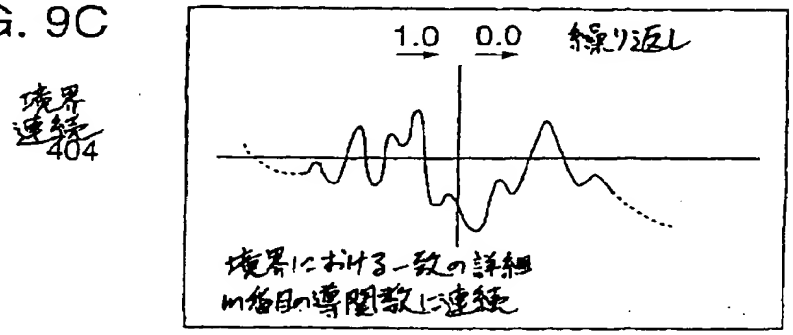
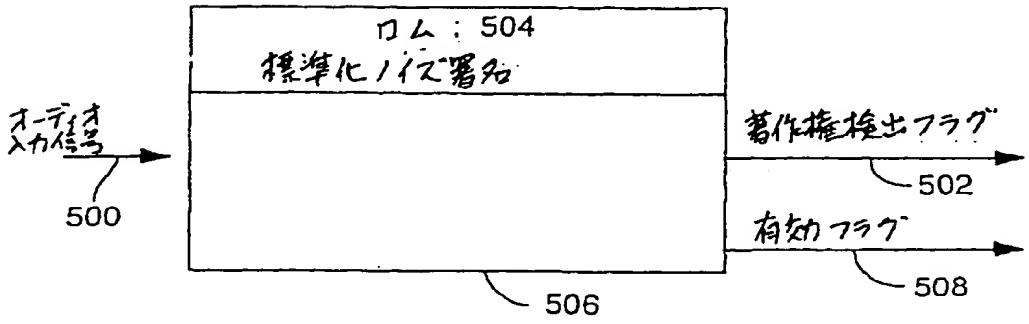


FIG. 9C



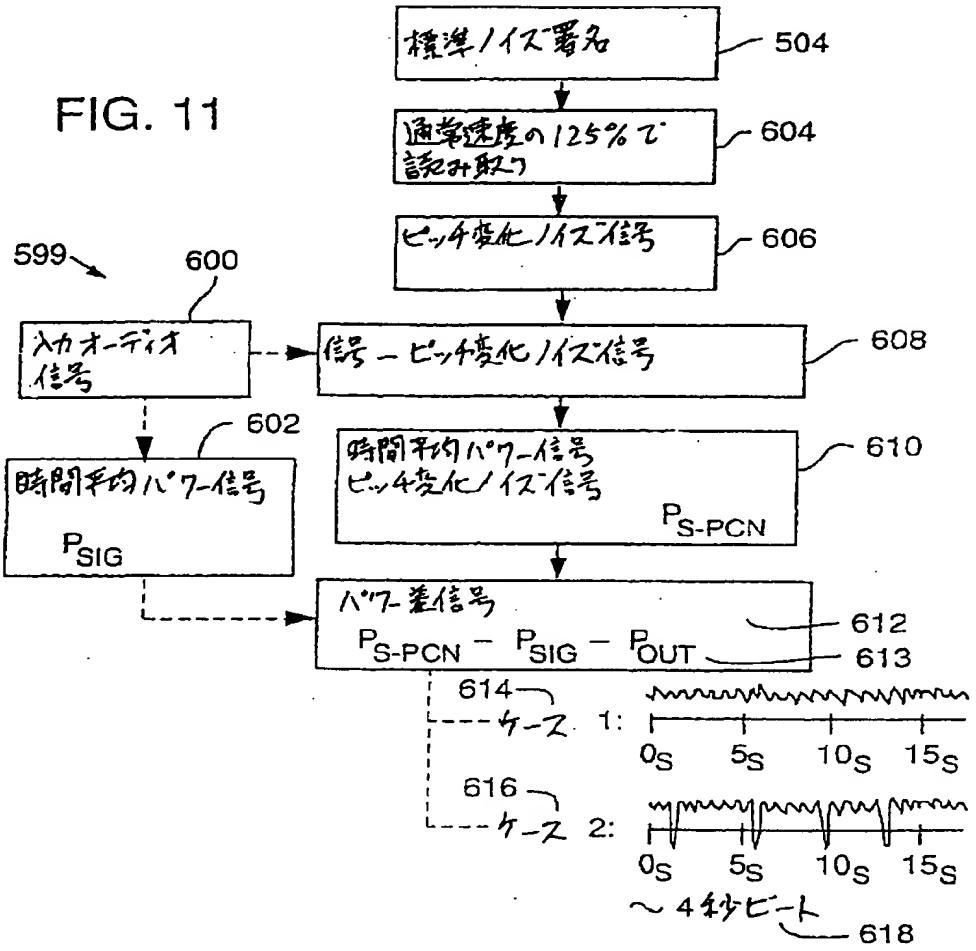
【図１０】

FIG. 10



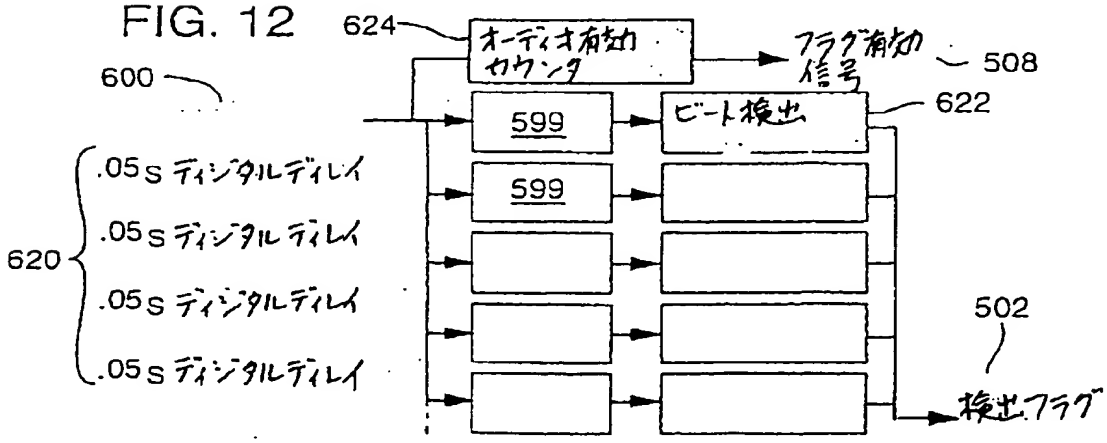
【図１１】

FIG. 11



【図１２】

FIG. 12



【国際調査報告】

INTERNATIONAL SEARCH REPORT

Inte: onal Application No
PCT/US 94/13366

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04B1/66 G11B20/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04B G11B G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	GB,A,2 196 167 (THORN EMI) 20 April 1988	1,2,5, 21,22
Y		3
A	see page 1, line 35 - page 2, line 35	4,9
Y	EP,A,0 411 232 (IBM) 6 February 1991 see page 4, line 7 - line 12 see page 5, line 28 - line 35	3
A	EP,A,0 372 601 (PHILIPS) 13 June 1990 see column 3, line 47 - column 4, line 12 see column 7, line 3 - line 17	1
A	DE,A,38 06 411 (DEUTSCHE THOMSON-BRANDT) 7 September 1989 see column 3, line 5 - column 4, line 25	1

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 May 1995	24. 05. 95

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Holper, G

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

.nformation on patent family members

Inter nal Application No
PCT/US 94/13366

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
GB-A-2196167	20-04-88	NONE		
EP-A-0411232	06-02-91	JP-A-	3165181	17-07-91
		US-A-	5204756	20-04-93
EP-A-0372601	13-06-90	NL-A-	8802769	01-06-90
		NL-A-	8901032	01-06-90
		AT-T-	118932	15-03-95
		AU-B-	626605	06-08-92
		AU-A-	4456889	31-05-90
		DE-D-	68921305	30-03-95
		JP-A-	2183468	18-07-90
		US-A-	5161210	03-11-92
DE-A-3806411	07-09-89	NONE		

Form PCT/ISA/210 (patent family annex) (July 1992)

フロントページの続き

(31)優先権主張番号　３２７，４２６
(32)優先日　1994年10月21日
(33)優先権主張国　米国（ＵＳ）
(81)指定国　ＥＰ(ＡＴ，ＢＥ，ＣＨ，ＤＥ，
ＤＫ，ＥＳ，ＦＲ，ＧＢ，ＧＲ，ＩＥ，ＩＴ，ＬＵ，Ｍ
Ｃ，ＮＬ，ＰＴ，ＳＥ)，ＣＡ，ＪＰ，ＵＳ